

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



STIC Search Report

EIC 2100

STIC Database Tracking Number: 125740

TO: Insun Kang
Location: 4A17
Art Unit : 2124
Monday, June 28, 2004

Case Serial Number: 09/873700

From: Geoffrey St. Leger
Location: EIC 2100
PK2-4B30
Phone: 308-7800

geoffrey.stleger@uspto.gov

Search Notes

Dear Examiner Kang,

Attached please find the results of your search request for application 09/873700. I searched Dialog's foreign patent files, technical databases, product announcement files and general files.

Please let me know if you have any questions.

Regards,

Geoffrey St. Leger
Geoffrey St. Leger
4B30/308-7800



STIC EIC 2100 125740 Search Request Form 125

Today's Date:

6/28/04

What date would you like to use to limit the search?

Priority Date:

Other:

Name Insun Kang

AU 2124 Examiner # 80013

Room # 4A17 Phone 305-6465

Serial # 09/873700

Format for Search Results (Circle One):

PAPER DISK EMAIL

Where have you searched so far?

USP DWPI EPO JPO ACM IBM TDB

IEEE INSPEC SPI Other _____

Is this a "Fast & Focused" Search Request? (Circle One) YES NO

A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at <http://ptoweb/patents/stic/stic-tc2100.htm>.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

Please call me when you are available.

STIC Searcher Geoffrey St. Leger Phone 308-7800

Date picked up 6/28/4 Date Completed 6/28/4



File 8: Ei Compendex(R) 1970-2004/Jun W3
 (c) 2004 Elsevier Eng. Info. Inc.
 File 35: Dissertation Abs Online 1861-2004/May
 (c) 2004 ProQuest Info&Learning
 File 202: Info. Sci. & Tech. Abs. 1966-2004/May 14
 (c) 2004 EBSCO Publishing
 File 65: Inside Conferences 1993-2004/Jun W3
 (c) 2004 BLDSC all rts. reserv.
 File 2: INSPEC 1969-2004/Jun W2
 (c) 2004 Institution of Electrical Engineers
 File 94: JICST-EPlus 1985-2004/May W5
 (c) 2004 Japan Science and Tech Corp(JST)
 File 483: Newspaper Abs Daily 1986-2004/Jun 24
 (c) 2004 ProQuest Info&Learning
 File 6: NTIS 1964-2004/Jun W4
 (c) 2004 NTIS, Intl Cpyrght All Rights Res
 File 144: Pascal 1973-2004/Jun W3
 (c) 2004 INIST/CNRS
 File 434: SciSearch(R) Cited Ref Sci 1974-1989/Dec
 (c) 1998 Inst for Sci Info
 File 34: SciSearch(R) Cited Ref Sci 1990-2004/Jun W3
 (c) 2004 Inst for Sci Info
 File 99: Wilson Appl. Sci & Tech Abs 1983-2004/May
 (c) 2004 The HW Wilson Co.
 File 583: Gale Group Globalbase(TM) 1986-2002/Dec 13
 (c) 2002 The Gale Group
 File 266: FEDRIP 2004/Apr
 Comp & dist by NTIS, Intl Copyright All Rights Res
 File 95: TEME-Technology & Management 1989-2004/Jun W1
 (c) 2004 FIZ TECHNIK
 File 438: Library Lit. & Info. Science 1984-2004/May
 (c) 2004 The HW Wilson Co
 File 104: AeroBase 1999-2004/May
 (c) 2004 Contains copyrighted material

Set	Items	Description
S1	1392130	INTERFACE? ? OR DECLARATION? ?
S2	1956183	CLASS OR CLASSES OR IMPLEMENTATION? ?
S3	2257	S1(10N)S2(10N) (ASSOCIAT? OR CORRELAT? OR MAP???? OR MATCH?- ?? OR RELATE? ? OR RELATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNECT??? OR AFFILIAT? OR BIND???)
S4	296	(SIGNATURE OR NAME)()MATCHING OR CALLING()CONTEXT
S5	1	DEFAULT(3N) (ASSOCIATION()RULE? ?)
S6	0	OVERRIDE(3N) (ASSOCIATION()RULE? ?)
S7	681306	MSIL OR IL OR INTERMEDIATE()LANGUAGE OR BYTECODE
S8	361272	RUNTIME OR RUN()TIME OR CL OR CLS
S9	194128	COMPIL?????
S10	168	S3 AND S4:S9
S11	92	S9 AND S10
S12	63	RD (unique items)
S13	766	(INTERFACE? ? OR DECLARATION? ?) (7N)IMPLEMENTATION(7N) (ASS- OCIAT? OR CORRELAT? OR MAP???? OR MATCH??? OR RELATE? ? OR RE- LATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNEC- T??? OR AFFILIAT? OR BIND???)
S14	21	S12 AND S13

14/5/1 (Item 1 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

05654727 E.I. No: EIP00095338227

Title: **Multimedia-editor for making findings in radiology**
Author: Vorwerk, Lutz; Meinel, Christoph
Corporate Source: Inst of Telematics, Trier, Ger
Conference Title: CBMS 2000: 13th IEEE Sympoisum on Computer-Based Medical Systems
Conference Location: Houston, TX, USA Conference Date: 19000622-19000624
Sponsor: IEEE
E.I. Conference No.: 57265
Source: Proceedings of the IEEE Symposium on Computer-Based Medical Systems 2000. IEEE, Los Alamitos, CA, USA. p 297-302
Publication Year: 2000
CODEN: PSCSFM ISSN: 1063-7125
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications)
Journal Announcement: 0010W4

Abstract: This summary describes the development of a multimedia-editor for radiology left bracket 1 right bracket which allows to load, to store and to work with radiological reports. These reports conform to the DICOM-standard by using the supplement 'structured reporting' left bracket 8 right bracket of DICOM left bracket 7 right bracket . The user-interface of the editor is implemented in Java. A DICOM toolkit is used to implement the structure of the reports. The toolkit is implemented in the programming language C and C plus plus and can be **compiled** on the operating systems Linux, Windows NT/9x, and Solaris. The programming languages used for the **implementation** of the user **interface** and the construction of DICOM-conform reports are **connected** via the Java native **interface** (JNI). A possibility to record spoken language or noises, to visualize DICOM images and to edit natural text is provided. (Author abstract) 9 Refs.

Descriptors: *Medical computing; Radiology; Multimedia systems; User interfaces; C (programming language); Java programming language

Identifiers: Multimedia editor; Java native interface

Classification Codes:

723.1.1 (Computer Programming Languages)

723.5 (Computer Applications); 461.1 (Biomedical Engineering); 461.6 (Medicine); 722.2 (Computer Peripheral Equipment); 723.1 (Computer Programming)

723 (Computer Software); 461 (Biotechnology); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING); 46 (BIOENGINEERING)

14/5/2 (Item 2 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

04602355 E.I. No: EIP97013495985

Title: **Communicating data-parallel tasks: An MPI library for HPF**
Author: Foster, Ian T.; Kohr, David R. Jr.; Krishnaiyer, Rakesh; Choudhary, Alok
Corporate Source: Argonne Natl Lab, Argonne, IL, USA
Conference Title: Proceedings of the 1996 3rd International Conference on High Performance Computing, HiPC
Conference Location: Trivandrum, India Conference Date: 19961219-19961222
Sponsor: IEEE; ACM SIGARCH
E.I. Conference No.: 45855
Source: Proceedings of the 1996 3rd International Conference on High Performance Computing, HiPC 1996. IEEE, Piscataway, NJ, USA, 96TB100074. p 433-438
Publication Year: 1996
CODEN: 002504
Language: English

Document Type: CA; (Conference Article) Treatment: G; (General Review)
Journal Announcement: 9703W2

Abstract: High Performance Fortran (HPF) has emerged as a standard dialect of Fortran for data-parallel computing. However, HPF does not support task parallelism or heterogeneous computing adequately. This paper presents a summary of our work on a library-based approach to support task parallelism, using MPI as a coordination layer for HPF. This library enables a wide variety of applications, such as multidisciplinary simulations and pipeline computations, to take advantage of combined task and data parallelism. An HPF **binding** for MPI raises several **interface** and communication issues. We discuss these issues and describe our **implementation** of an HPF/MPI library that operates with a commercial HPF **compiler**. We also evaluate the performance of our library using a synthetic communication benchmark and a multiblock application. (Author abstract) 7 Refs.

Descriptors: Parallel processing systems; FORTRAN (programming language); Pipeline processing systems; Computer simulation; Interfaces (computer); Program **compilers**; Data communication systems

Identifiers: Multiblock application; Synthetic communication benchmark

Classification Codes:

723.1.1 (Computer Programming Languages)
722.4 (Digital Computers & Systems); 723.1 (Computer Programming);
723.5 (Computer Applications); 722.2 (Computer Peripheral Equipment)
722 (Computer Hardware); 723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

14/5/3 (Item 3 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

04557616 E.I. No: EIP96110414607

Title: Machine-independent debugger

Author: Hanson, David R.; Raghavachari, Mukund

Corporate Source: Princeton Univ, Princeton, NJ, USA

Source: Software - Practice and Experience v 26 n 11 Nov 1996. p 1277-1299

Publication Year: 1996

CODEN: SPEXBL ISSN: 0038-0644

Language: English

Document Type: JA; (Journal Article) Treatment: G; (General Review)

Journal Announcement: 9701W2

Abstract: cdb is a simple, source-level debugger for ANSI/ISO C programs **compiled** by lcc, a retargetable C **compiler**. cdb is completely independent of its target architecture. This independence is achieved by embedding a small amount of itself - a 'hub' - in the program to be debugged and by having the **compiler** emit machine-independent symbol-table data and **related** code. This paper describes the design of a nub **interface** for debugging, a target-independent **implementation** of this **interface** for cdb, and the **implementation** of cdb itself. When compared to platform-specific debuggers, cdb's **implementation** is minuscule: the nub, debugger, and **associated** communications routines take about 1500 lines of C, and fewer than 600 lines were added to lcc's front end to emit symbol tables and breakpoint hooks. All this code is machine dependent. (Author abstract) 19 Refs.

Descriptors: Program debugging; Program **compilers**; Interfaces (computer); Computer architecture; Codes (symbols); Computer operating systems

Identifiers: Source level debuggers; Lines of code (LOC); Nub interface; Breakpoint hooks

Classification Codes:

723.1 (Computer Programming); 722.1 (Data Storage, Equipment & Techniques); 723.2 (Data Processing)
723 (Computer Software); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

14/5/4 (Item 4 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

04229403 E.I. No: EIP95082821465

Title: Fully conformant implementation of ECMA-162

Author: Stuckey, Richard

Corporate Source: ICL Enterprises, Reading, Engl

Source: Ada User Journal v 16 n 2 Jun 1995. p 83-94

Publication Year: 1995

CODEN: AUJOET ISSN: 0268-652X

Language: English

Document Type: JA; (Journal Article) Treatment: G; (General Review)

Journal Announcement: 9510W3

Abstract: ICL has developed a portable **implementation** of the Ada **interfaces** to PCTE as specified by ECMA-162. The **interfaces** **map** the functionality required onto that provided by the C interfaces to PCTE as specified by ECMA-158. The process of implementing the interfaces revealed a number of errors in the ECMA PCTE standards, such as errors in ECMA-162 concerning the mapping of ECMA-149 onto Ada, errors in ECMA-158 such as missing operations or functions with incorrect parameter modes, discrepancies between the Ada and C bindings and errors in ECMA-149. The architecture of the interfaces and their test harness has been designed to allow easy porting from one PCTE implementation to another, and also from one Ada **compilation** system to another; some major constraints were imposed by the use of the C interfaces as the underlying platform, particularly regarding Ada's multi-threading abilities. The advantages of using the interfaces include the benefits of being able to implement tools in Ada instead of C; insulation from the underlying PCTE implementation; and the provision of facilities (e.g. call tracing) between tools and PCTE. (Author abstract) 11 Refs.

Descriptors: *Ada (programming language); User interfaces; Errors; Standards; C (programming language); Computer aided software engineering; Codes (symbols)

Identifiers: Ada interfaces

Classification Codes:

723.1.1 (Computer Programming Languages)

723.1 (Computer Programming); 722.2 (Computer Peripheral Equipment);

902.2 (Codes & Standards); 723.5 (Computer Applications)

723 (Computer Software); 722 (Computer Hardware); 902 (Engineering Graphics & Standards)

72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING)

14/5/5 (Item 5 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

02824716 E.I. Monthly No: EIM8911-044714

Title: On generalizations in networking software to encourage code portability.

Author: Peters, Patrick; Deruz, Roy; Sung, Chiun-Teh; Wang, Christine; Meandzija, Branislav

Corporate Source: Texas Instrum, Dallas, TX, USA

Conference Title: Eighth Annual Joint Conference of the IEEE Computer and Communications Societies - Technology: Emerging or Converging?

Conference Location: Ottawa, Ont., Canada Conference Date: 19890423

E.I. Conference No.: 12403

Source: Proceedings - IEEE INFOCOM v I (of 3). Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat n 89CH2702-9), Piscataway, NJ, USA. p 261-267

Publication Year: 1989

CODEN: PINFEZ ISSN: 0743-166X

Language: English

Document Type: PA; (Conference Paper) Treatment: T; (Theoretical)

Journal Announcement: 8911

Abstract: The principal issues involved in porting networking software are discussed and solutions that have been used to implement a standard environment interface to encourage code portability are reported. This

interface has been designed to provide a uniform environment to protocol drivers generated by the Archetype language **compiler**. The structure of the present **implementation** is outlined and issues **relating** to the environment **interface** are elaborated. The **implementation** provides both portable data representation and portable systems services. 19 refs.

Descriptors: *COMPUTER NETWORKS; COMPUTER SOFTWARE--Portability; COMPUTER OPERATING SYSTEMS; DIGITAL COMMUNICATION SYSTEMS

Identifiers: NETWORKING SOFTWARE; ARCHETYPE LANGUAGE **COMPILER**; PORTABLE DATA REPRESENTATION; BERKELEY UNIX; DIGITAL VMS OPERATING SYSTEMS

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

14/5/6 (Item 6 from file: 8)

DIALOG(R)File 8:EI Compendex(R)

(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

02293567 E.I. Monthly No: EI8707066894

Title: **LIDO - A SILICON COMPILER PREPROCESSOR.**

Author: Przemyslaw, Bakowski; Adam, Pawlak

Corporate Source: Univ des Sciences et Techniques de Lille, Fr

Source: Microprocessing and Microprogramming v 20 n 1-3 Apr 1987, Short Notes, Euromicro '86, Venice, Italy, 1986 p 167-171

Publication Year: 1987

CODEN: MMICDT

Language: ENGLISH

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical).

Journal Announcement: 8707

Abstract: This paper presents the main features of LIDO project aimed to develop VLSI and ULSI original architectures at register transfer level. LIDO project consists of LIDO language and LIDO interpretation system. Both have been designed to operate as silicon **compiler** preprocessor. LIDO system assures the functional analysis of the designs described at functional register transfer (RT) level as well as at structural RT level. LIDO language provides a large design exploration space including multiprocessing, data flow, virtual memory, and other architectures. LIDO constructs are closely **related** to highly structured **implementation** blocks, e. g. , PLAs, SLAs. This feature permits creating an efficient **interface** between structural RT level and the **corresponding** silicon topology. LIDO project has been designed to operate in UNIX environment. LIDO language symbolics is C-like symbolics. LIDO system has been implemented in C language. The LIDO system commands are UNIX-like commands. (Edited author abstract) 19 refs.

Descriptors: COMPUTER OPERATING SYSTEMS--*Program **Compilers**; COMPUTER PROGRAMMING LANGUAGES

Identifiers: SILICON **COMPILER**; PREPROCESSOR; VLSI AND ULSI ORIGINAL ARCHITECTURES; REGISTER TRANSFER; LIDO LANGUAGE

Classification Codes:

722 (Computer Hardware); 723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

14/5/7 (Item 7 from file: 8)

DIALOG(R)File 8:EI Compendex(R)

(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

00993765 E.I. Monthly No: EI8102011894 E.I. Yearly No: EI81018037

Title: **SEPARATE COMPILATION AND PARTIAL SPECIFICATION IN PASCAL.**

Author: Celentano, Augusto; Vigna, Pierluigi Della; Ghezzi, Carlo; Mandroili, Dino

Corporate Source: Politec di Milano, Italy

Source: IEEE Transactions on Software Engineering v SE-6 n 4 Jul 1980 p 320-328

Publication Year: 1980

CODEN: IESEDJ ISSN: 0098-5589

Language: ENGLISH

Journal Announcement: 8102

Abstract: A separate **compilation** scheme for Pascal is proposed which couples protection and **compile** -time checking with easy reusability of modules in different systems. The proposal includes a partial specification facility for module **interfaces** which requires a **linking** procedure more elaborate than in conventional **linkers**. **Implementation** issues of the proposal are also discussed. 20 refs.

Descriptors: COMPUTER PROGRAMMING LANGUAGES--*PASCAL; COMPUTER OPERATING SYSTEMS--Program **Compilers**

Classification Codes:

7.2 (Computer Software)

7.1 (COMPUTERS & DATA PROCESSING)

14/5/8 (Item 1 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2004 ProQuest Info&Learning. All rts. reserv.

01825295 ORDER NO: AADAA-I3007184

Object-based message passing in high performance computing using Java

Author: Ko, Sung-Hoon

Degree: Ph.D.

Year: 2000

Corporate Source/Institution: Syracuse University (0659)

Adviser: Geoffrey C. Fox

Source: VOLUME 62/03-B OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 1464. 139 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

ISBN: 0-493-16510-X

We present designs for Java **interfaces** to High Performance Computing softwares, and research **implementation** issues **associated** with their development. We investigate various issues and options for parallel programming in Java—data parallelism and message passing libraries. We discuss motivations for introducing HPJava, an HPspmd programming model.

As part of the larger environment of HPJava, we have designed and developed mpiJava, an object-oriented Java interface to MPI. This has been implemented by "native methods" wrappers around some pre-existing MPI implementation. We describe the design of the mpiJava API and the issues associated with its development. We present and then discuss performance measurements made of communications bandwidth and latency to compare mpiJava with C and Fortran bindings of MPI. We discuss adoption of the Java object serialization model in mpiJava for marshalling general communication data in MPI-like APIs. This approach is compared with a Java transcription of the standard MPI derived datatype mechanism. We evaluate overheads introduced by object serialization in mpiJava, then present optimized methods that reduce serialization overheads. We present graphical mpiJava Potts model simulation and evaluate the performance of Ising model simulation using Metropolis and Swendsen-Wang cluster algorithms. Benchmark results are compared with native parallel and sequential codes.

We have publically released mpiJava on a Web site. This includes complete source, makefiles, configuration scripts, **compiled** libraries for WMPI, test codes (based on the IBM MPI test suite), example applications, javadoc documentation, and installation and usage notes.

14/5/9 (Item 2 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2004 ProQuest Info&Learning. All rts. reserv.

01650105 ORDER NO: AADNQ-27922

MEMORY-SYSTEM DESIGN CONSIDERATIONS FOR DYNAMICALLY-SCHEDULED MICROPROCESSORS (REGISTER FILE, DATA CACHE, STREAM BUFFERS)

Author: FARKAS, KEITH ISTVAN

Degree: PH.D.

Year: 1997

Corporate Source/Institution: UNIVERSITY OF TORONTO (CANADA) (0779)

Advisers: PAUL CHOW; ZVONKO VRANESIC
Source: VOLUME 59/06-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 2933. 199 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL
Descriptor Codes: 0544
ISBN: 0-612-27922-7

Dynamically-scheduled processors challenge hardware and software architects to develop designs that balance hardware complexity and **compiler** technology against performance targets. This dissertation presents a first thorough look at some of the issues introduced by this hardware complexity. The focus of the investigation of these issues is the register file and the other components of the data memory system. These components are: the lockup-free data cache, the stream buffers, and the interface to the lower levels of the memory system.

The investigation is based on software models. These models incorporate the features of a dynamically-scheduled processor that affect the design of the data-memory components. The models represent a balance between accuracy and generality, and are used to identify performance trends and design relationships rather than to evaluate the performance of specific designs.

The models are used to identify and explore the relationships between the following system aspects: the dispatch queue, the physical registers, the type of exceptions, the issue width of the processor and the number of functional units, the lockup-free data cache, the stream buffers, and the **interface** to the lower levels of the memory system. **Implementation - related relationships** are exposed through an analysis of how various design decisions impact the cycle time of a register file.

Following from this analysis, an investigation of a novel architecture is conducted. The novelty of this architecture is the partitioning of key components to reduce the cycle time of the processor. A comprehensive examination is given of the software and hardware complexities the partitioning introduces. Static instruction scheduling algorithms are developed that address these complexities. Evaluation of these algorithms indicates that for the configurations considered, the partitioned architecture is competitive, and warrants further investigation.

Finally, further insights into the novel architecture and dynamically-scheduled processors in general are developed through an analysis of the non-determinism introduced by dynamic scheduling, and the challenges this factor presents to computer architects who must design systems that can be used to full advantage by software. It is shown that static instruction scheduling for dynamically-scheduled processors is unsolvable with known techniques.

14/5/10 (Item 3 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online
(c) 2004 ProQuest Info&Learning. All rts. reserv.

01474869 ORDER NO: AADAA-I9611552
A RETARGETABLE, OPTIMIZING LINKER (PARTIAL EVALUATION, MLD)
Author: FERNANDEZ, MARIA FRANCESCA
Degree: PH.D.
Year: 1996
Corporate Source/Institution: PRINCETON UNIVERSITY (0181)
Source: VOLUME 56/12-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 6858. 132 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

Development of reusable software is an important software-engineering problem. Researchers address this problem by designing programming languages with features to support the development of modular software. Object-oriented languages support modularity by providing abstractions, such as Modula-3's abstract data types, that separate a software component's **interface** from its **implementation**. Some object-oriented features, however, incur **runtime** costs, because they shift **binding** time, which is the time when the complete representations of types and

variables are known, from **compile** time to **run** time . Late binding makes it difficult for a **compiler** to implement these features efficiently.

This thesis describes a software approach to the problem of implementing high-level programming languages with late binding. We show how Modula-3's and C\sp{++}\$'s features that require late binding can be implemented more efficiently with an optimizing linker. We describe the design and implementation of mld, a retargetable, optimizing linker, for the MIPS, SPARC, and Intel 486 architectures, mld's optimizations permit efficient implementation of Modula-3's module-encapsulation and data-abstraction features, we present the results of applying mld's simple optimization techniques to several Modula-3 programs and report their execution times on the MIPS and Intel 486.

mld's optimizations use information--the complete type hierarchy, external procedures, and profiles--that is usually unavailable until link time and therefore not available to an optimizing **compiler** . The optimizations simplicity contributes to their effectiveness and efficiency, because they do not require complex algorithms, and they are easy to implement. Most important, mld's optimizations reduce the **runtime** costs of strong encapsulation features without compromising program modularity.

Unlike conventional linkers, mld links mill, a machine-independent intermediate code, instead of machine-dependent object code. Linking intermediate code simplifies mld's implementation, because mill is a natural representation for applying mld's machine-independent optimizations and permits optimization of procedures in libraries as well as those in applications. Linking intermediate code is expensive, because code generation is delayed until link time. mld offsets this cost by using fast code-generation technology. We report the costs of using mld and describe the software technology that helps makes link-time code generation feasible on multiple targets.

14/5/11 (Item 4 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2004 ProQuest Info&Learning. All rts. reserv.

01320900 ORDER NO: NOT AVAILABLE FROM UNIVERSITY MICROFILMS INT'L.

MATCHING OF VISION AND CAD MODELS: APPLICATION TO VISION SYSTEMS OFF-LINE LEARNING IN A CIM CONTEXT

Original Title: MISE EN CORRESPONDANCE DE MODELES VISION ET CAO:

APPLICATION A L'APPRENTISSAGE HORS-LIGNE DES SYSTEMES DE VISION DANS UN CONTEXTE CIM

Author: LUCAS, YVES

Degree: DR.

Year: 1993

Corporate Source/Institution: INSTITUT NATIONAL DES SCIENCES APPLIQUES
DE LYON (FRANCE) (5285)

Source: VOLUME 54/04-C OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 1254. 186 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

Language: FRENCH

Off-line programming of production tools from CAD/CAM data is a great step toward their integration according to a CIM context. The CAD-vision connection developed in this thesis extends to vision systems this programming mode already used to suppress manual learning on NC-machines and some industrial robots. This work includes a theoretical specification of the **connection** followed by a practical **implementation** . The CAD-vision **connection** is studied on three levels: models to **match** , hardware to **interface** and tasks to program. Concerning the models, we propose a view-dependent description, free of projective simplifications, based on spatial outlines. It combines a discriminating parametric description with a local description ensuring hypothesis consistency. The CAD model preprocessing outputs a **compiled** , vision oriented description, which makes matching easier. On the material level, we describe several programming modes, replacing samples learning, using postprocessors,

standard formats and artificial images. Finally, we delimit the application field of the CAD-vision connection, brought into operation on recognition and reconstruction tasks.

14/5/12 (Item 1 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

6765374 INSPEC Abstract Number: C2001-01-6110P-002

Title: **Compositional development of performance models in POEMS**

Author(s): Browne, J.C.; Berger, E.; Dube, A.

Author Affiliation: Dept. of Comput. Sci., Texas Univ., Austin, TX, USA

Journal: International Journal of High Performance Computing Applications
vol.14, no.4 p.283-91

Publisher: Sage Science Press,

Publication Date: Winter 2000 Country of Publication: USA

CODEN: IHPCFL ISSN: 1078-3482

SICI: 1078-3482(200024)14:4L.283:CDPM;1-I

Material Identity Number: G357-2000-004

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: Performance models are software systems in which the components implement abstractions of the behavior of a total system. The paper describes a capability for semiautomatic development of performance models of computer systems spanning applications, operating systems, and hardware by composition from a library of components. Compositional development of performance models is a domain-specific instance of the general problem of software component reuse or design reuse. The concepts enabling compositional development of performance models in the performance-oriented end-to-end modelling system (POEMS) are encapsulation of analysis-level objects with associative interfaces and hierarchical dynamic data flow graphs as a structuring model. Objects with associative interfaces are called compositional objects. Compositional objects and hierarchical dynamic data flow graphs provide a framework for the development of performance models that incorporate multiple modes of evaluation, span multiple semantic domains, span multiple levels of abstraction, and parallel **implementation**. Algorithms for composition through **associative interfaces** with automatic generation of parallel executables for the performance models are defined. (17 Refs)

Subfile: C

Descriptors: data encapsulation; data flow graphs; parallel programming; program **compilers**; software performance evaluation; software reusability; specification languages

Identifiers: compositional development; performance models; POEMS; software systems; semiautomatic development; software component reuse; design reuse; performance-oriented end-to-end modelling system; analysis-level objects; hierarchical dynamic data flow graphs; structuring model; associative interfaces; compositional objects; abstraction; parallel implementation

Class Codes: C6110P (Parallel programming); C6110J (Object-oriented programming); C1160 (Combinatorial mathematics); C6110B (Software engineering techniques); C6150C (Compilers, interpreters and other processors); C6140D (High level languages)

Copyright 2000, IEE

14/5/13 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

6034930 INSPEC Abstract Number: C9811-6140D-018

Title: **VISCO: bringing visual spatial querying to reality**

Author(s): Wessel, M.; Haarslev, V.

Author Affiliation: Dept. of Comput. Sci., Hamburg Univ., Germany

Conference Title: Proceedings. 1998 IEEE Symposium on Visual Languages
(Cat. No.98TB100254) p.170-7

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA

Publication Date: 1998 Country of Publication: USA x+307 pp.
ISBN: 0 8186 8712 6 Material Identity Number: XX98-02488
U.S. Copyright Clearance Center Code: 0 8186 8712 6/98/\$10.00
Conference Title: Proceedings 1998 IEEE Symposium on Visual Languages
Conference Sponsor: IEEE Comput. Soc. Tech. Committee on Multimedia
Comput

Conference Date: 1-4 Sept. 1998 Conference Location: Halifax, NS,
Canada

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: The paper reports on the evolution of the spatial (sketch based) query language VISCO and its implementation. The first design of VISCO's query language was presented at VL '97. The language is based on a strong naive physics metaphor for query objects (e.g. marbles, nails, rubber bands). We review the prominent aspects of the revised version of VISCO's query language. The main focus of the paper is on VISCO's **implementation** using city **maps** of Hamburg as example domain. Its innovative user **interface** consists of three interconnected components: a graphical (syntax directed) query editor and visual language **compiler**, a browser for inspecting the query results, and a map viewer for browsing the spatial database. We also briefly report on the process of **compiling**, optimizing, and executing VISCO's queries. (7 Refs)

Subfile: C

Descriptors: graphical user interfaces; program **compilers**; query languages; visual databases; visual languages; visual programming

Identifiers: VISCO; visual spatial querying; sketch based query language; query language; naive physics metaphor; query objects; city maps; Hamburg; innovative user interface; interconnected components; graphical query editor; visual language **compiler**; browser; map viewer; spatial database; query processing

Class Codes: C6140D (High level languages); C6110V (Visual programming); C6130B (Graphics techniques); C6180G (Graphical user interfaces); C6150C (Compilers, interpreters and other processors); C6160S (Spatial and pictorial databases)

Copyright 1998, IEE

14/5/14 (Item 3 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

5862157 INSPEC Abstract Number: C9804-6115-037

Title: **Visual programming in Smalltalk**

Author(s): Giffen, R.; Tomek, I.

Author Affiliation: Jodrey Sch. of Comput. Sci., Acadia Univ., Wolfville, NS, Canada

Conference Title: Educational Multimedia and Hypermedia, 1996. Proceedings of ED-MEDIA 96 - World Conference on Educational Multimedia and Hypermedia p.772

Editor(s): Carlson, P.; Makedon, F.

Publisher: Assoc. Adv. Comput. Educ, Charlottesville, VA, USA

Publication Date: 1996 Country of Publication: USA xvi+879 pp.

ISBN: 1 880094 21 5 Material Identity Number: XX96-01492

Conference Title: Proceedings of ED-MEDIA 96 World Conference on Educational Multimedia and Hypermedia

Conference Date: 17-22 June 1996 Conference Location: Boston, MA, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: Summary form only given. With the shift towards multimedia attempts have been made to test whether visually based software development environments would have advantages over conventional text based environments. We selected Smalltalk to implement a new type of such an alternative environment. Our ultimate goal is to test what advantages a graphically based environment might have over a textual one. The reasons why we selected Smalltalk for the experiment are its uncompromising object orientation, availability of all source code within the environment, extreme ease of experimentation due to the modifiability and extendibility of its library, and the fact that Smalltalk does not require lengthy

compilation and linking during application development. The conventional Smalltalk program development interface is textual but implementation of an alternative GUI is relatively easy. The visual browser was developed as a part of an MSc thesis. Two radio buttons have been added to the standard browser to make it possible to select either the conventional textual representation or the graphical representation, and switching from one mode to another automatically changes the display. The format used for display is independent of the style used to develop the program. (0 Refs)

Subfile: C

Descriptors: graphical user interfaces; programming environments; Smalltalk; visual programming

Identifiers: visual programming; Smalltalk; multimedia; visually based software development environments; text based environments; object orientation; program development interface; GUI; visual browser; radio buttons

Class Codes: C6115 (Programming support); C6110V (Visual programming); C6180G (Graphical user interfaces)

Copyright 1998, IEE

14/5/15 (Item 4 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

5592370 INSPEC Abstract Number: C9707-6110J-011

Title: **Literate programming and code reuse**

Author(s): Hendseth, S.

Author Affiliation: TTS Autom., Bergen, Norway

Journal: Dr. Dobb's Journal vol.22, no.6 p.18-20, 22, 24, 97-8

Publisher: Miller Freeman,

Publication Date: June 1997 Country of Publication: USA

CODEN: DDJSDM ISSN: 1044-789X

SICI: 1044-789X(199706)22:6L:18:LPCR;1-S

Material Identity Number: B719-97005

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: The authors looks at ways in which literate programming helps you recognise the layout of a program. When given the freedom to structure program layout as you wish, you tend to group parts of the code differently than as dictated by the language or **compiler**. Rather than grouping all public-function **declarations** of a C++ **class** together in the public section of the **class**, for instance, you might group each function **declaration** with the **corresponding** function **implementation**. The author also examines how some of these new groups of code can be parametrised into reusable macros with the help of a simple macro processor. Specifically, the author shows how the application of design patterns such as Singleton and State can be supported by such macros. (0

Refs)

Subfile: C

Descriptors: C listings; macros; object-oriented programming; software reusability; structured programming; system documentation

Identifiers: code reuse; literate programming; program layout recognition; program layout structuring; public-function declarations; C++ class; reusable macros; macro processor; design patterns

Class Codes: C6110J (Object-oriented programming); C6110B (Software engineering techniques)

Copyright 1997, IEE

14/5/16 (Item 5 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

94169356 INSPEC Abstract Number: C9207-7420D-038

Title: **An interactive environment for real time implementation of control systems**

Author(s): Dahl, O.

Author Affiliation: Dept. of Autom. Control, Lund Inst. of Technol., Sweden

Conference Title: Computer Aided Design in Control Systems. Selected Papers from the IFAC Symposium p.65-70

Editor(s): Barker, H.A.

Publisher: Pergamon, Oxford, UK

Publication Date: 1992 Country of Publication: UK xiv+457 pp.

ISBN: 0 08 041269 6

Conference Sponsor: IFAC; IMACS

Conference Date: 15-17 July 1991 Conference Location: Swansea, UK

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: An approach to efficient implementation of real time control systems is presented. A **compiler** for translation of control algorithms is used in combination with a general program for real time control. The **compiler** translates control algorithms, written in a design language, to an **implementation** language, and generates code for **connecting** the control algorithms to the user **interface**. The translated algorithms are then automatically incorporated in the real time control program. The resulting executable program has a number of interactive facilities such as interconnection of controllers, plotting and textual display of all variables, and data logging. The design language that was chosen is Simnon, a language for simulation of nonlinear systems, and the implementation language that was chosen is Modula-2. The system has been used in research and education, and has reduced the implementation time considerably, e.g. when developing new laboratory exercises, or when a control algorithm is tested in a laboratory experiment. (10 Refs)

Subfile: C

Descriptors: control system CAD; program **compilers** ; real-time systems

Identifiers: real time control systems; **compiler** ; design language; implementation language; user interface; Simnon; nonlinear systems; Modula-2; laboratory exercises

Class Codes: C7420D (Control system design and analysis); C6150C (Compilers, interpreters and other processors)

14/5/17 (Item 6 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2004 Institution of Electrical Engineers. All rts. reserv.

03819519 INSPEC Abstract Number: C91017810

Title: Development of the knowledge based CAPP-interface linking CAD and CAM

Author(s): Sluga, A.; Peklenik, J.

Author Affiliation: Dept. of Control & Manuf. Syst., Ljubljana Univ., Yugoslavia

Conference Title: Software for Factory Automation. Proceedings of the IFIP TC 5/WG 5.3/IFORS Working Conference p.139-52

Editor(s): Sata, T.; Olling, G.

Publisher: North-Holland, Amsterdam, Netherlands

Publication Date: 1989 Country of Publication: Netherlands x+363 pp.

ISBN: 0 444 88046 1

Conference Sponsor: IFIP

Conference Date: 19-21 Oct. 1987 Conference Location: Tokyo, Japan

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: An efficient **implementation** of the computer-aided design (CAD) and manufacturing (CAM) requires a CAPP- **interface linking** and integrating both processes: the generation of information resulting in technical drawing and the generation of hardware resulting in machined parts. The investigation is based on the binary matrix (BCM) containing the geometrical primitives and the structure of the EXAPT-programming system, considering the machining database. The proposed generative process planning system (GPPS) uses expert system methodology, and is structured on the principle of complexity decomposition of planning operations. Particular decision levels of planning are related to various levels of abstraction and are illustrated in the GPPS scheme. An example of the decision making process for operation planning is shown. The expert system

interprets the technological knowledge with regard to the input data of the workpiece and on the basis of knowledge and data **compiled** in the technological D+K base. The knowledge is expressed in rule form and the programming procedure implemented in PROLOG. The modularity and clarity of the organization of technological knowledge and data will enable the system growth, flexibility, as well as its adaptability to specific manufacturing environments. (18 Refs)

Subfile: C

Descriptors: CAD/CAM; knowledge based systems; production control

Identifiers: knowledge based CAPP-interface; CAD; CAM; geometrical primitives; EXAPT-programming system; machining database; generative process planning system; expert system; decision making process; operation planning; D+K base; rule form; PROLOG

Class Codes: C7160 (Manufacturing and industry); C6170 (Expert systems); C7400 (Engineering)

14/5/18 (Item 1 from file: 6)

DIALOG(R)File 6:NTIS

(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

1794899 NTIS Accession Number: N94-22516/6

Development of the FITS Tools Package for Multiple Software Environments
(Abstract Only)

Pence, W. D. ; Blackburn, J. K.

National Aeronautics and Space Administration, Greenbelt, MD. Goddard Space Flight Center.

Corp. Source Codes: 013129001; NC999967

1992 1p

Languages: English

Journal Announcement: GRAI9411; STAR3205

In NASA, Washington, Second Annual Conference on Astronomical Data Analysis Software and Systems. Abstracts p 85.

NTIS Prices: (Order as N94-22438/3, PC A07/MF A02)

Country of Publication: United States

The HEASARC is developing a package of general purpose software for analyzing data files in FITS format. This paper describes the design philosophy which makes the software both machine-independent (it runs on VAXs, Suns, and DEC-stations) and software environment-independent. Currently the software can be **compiled** and linked to produce IRAF tasks, or alternatively, the same source code can be used to generate stand-alone tasks using one of two implementations of a user-parameter interface library. The machine independence of the software is achieved by writing the source code in ANSI standard Fortran or C, using the machine-independent FITSIO subroutine interface for all data file I/O, and using a standard user-parameter subroutine interface for all user I/O. The latter interface is based on the Fortran IRAF Parameter File **interface** developed at STScI. The IRAF tasks are built by **linking** to the IRAF **implementation** of this parameter **interface** library. Two other **implementations** of this parameter **interface** library, which have no IRAF dependencies, are now available which can be used to generate stand-alone executable tasks. These stand-alone tasks can simply be executed from the machine operating system prompt either by supplying all the task parameters on the command line or by entering the task name after which the user will be prompted for any required parameters. A first release of this FTOOLS package is now publicly available. The currently available tasks are described, along with instructions on how to obtain a copy of the software.

Descriptors: *Computer programming; *Software tools; *Subroutine libraries (Computers); Input/output routines; Machine-independent programs; Programming environments; C (Programming language); Data reduction; Fortran; Source programs; Subroutines

Identifiers: NTISNASA

Section Headings: 54C (Astronomy and Astrophysics--Astrophysics); 54B (Astronomy and Astrophysics--Astronomy and Celestial Mechanics); 62B (Computers, Control, and Information Theory--Computer Software)

14/5/19 (Item 2 from file: 6)

DIALOG(R)File 6:NTIS

(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

1400068 NTIS Accession Number: AD-A198 677/7

Analysis of Applications for an Interactive Maintenance-Aiding System.

Executive Summary

(Final rept. 1 Apr-1 Aug 8)

Modrick, J. A. ; Thomas, C. E. ; Strybel, T.

Battelle Columbus Labs., Research Triangle Park, NC.

Corp. Source Codes: 073414000; 412309

Sep 85 13p

Languages: English

Journal Announcement: GRAI8902

Prepared in cooperation with Honeywell Systems and Research Center, Minneapolis, Mn., Rept. no. 85SRC27. See also AD-A198 676.

Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.

NTIS Prices: PC A03/MF A01

Country of Publication: United States

Contract No.: DAAG29-81-D-0100

The objectives of the study were to 1) explore the feasibility and benefits of incorporating interactive job-aiding techniques into an interface to electronic documentation for depot-level maintenance tasks, 2) identify candidate job tasks for electronic aiding, 3) develop an initial concept for an interactive maintenance-aiding system (IMAS), and 4) develop a roadmap for **implementation** of an IMAS, including prototype development. The **relationship** of the proposed **interface**-oriented IMAS to the paper-oriented Automated Technical Order System (ATOS) is discussed. It is recommended that a menu-based authoring system for creating state-of-the-art electronic aiding interfaces, using ATOS and related files as input, be developed. The authoring system would allow depot maintenance managers and supervisors to compose, **compile**, and update maintenance-aiding materials to be used for interactive presentation to the technician. Prototype development and implementation plans are presented for two application areas: the Precision Measurement and Equipment Lab (PMEL) and F-16 checkout. Keywords: Job performance aiding, Electronic documentation, Authoring systems. (sdw)

Descriptors: *Job analysis; *Maintenance personnel; *Maintenance; *Supervisors; *Supply depots; Electronic equipment; Interfaces; Jobs; Laboratories; Measurement; Precision; Prototypes

Identifiers: *Job aids; Job performance aids; NTISDODXA

Section Headings: 70A (Administration and Management--Inventory Control); 70D (Administration and Management--Personnel Management, Labor Relations, and Manpower Studies); 92A (Behavior and Society--Job Training and Career Development); 74E (Military Sciences--Logistics, Military Facilities, and Supplies)

14/5/20 (Item 3 from file: 6)

DIALOG(R)File 6:NTIS

(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

1400067 NTIS Accession Number: AD-A198 676/9

Analysis of Applications for an Interactive Maintenance-Aiding System

(Final rept. 1 Apr-1 Aug 8)

Modrick, J. A. ; Thomas, C. E. ; Strybel, T.

Battelle Columbus Labs., Research Triangle Park, NC.

Corp. Source Codes: 073414000; 412309

Sep 85 79p

Languages: English

Journal Announcement: GRAI8902

Prepared in cooperation with Honeywell Systems and Research Center, Minneapolis, Mn., Rept. no. 85SRC27. See also AD-A198 677.

Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road,

Springfield, VA, 22161, USA.

NTIS Prices: PC A05/MF A01

Country of Publication: United States

Contract No.: DAAG29-81-D-0100

The objectives of the study were to 1) explore the feasibility and benefits of incorporating interactive job-aiding techniques into an interface to electronic documentation for depot-level maintenance tasks, 2) identify candidate job tasks for electronic aiding, 3) develop an initial concept for an interactive maintenance-aiding system (IMAS), and 4) develop a roadmap for **implementation** of an IMAS, including prototype development. The **relationship** of the proposed **interface** -oriented IMAS to the paper-oriented Automated Technical Order System (ATOS) is discussed. It is recommended that a menu-based authoring system for creating state-of-the-art electronic aiding interfaces, using ATOS and related files as input, be developed. This authoring system would allow depot maintenance managers and supervisors to compose, **compile**, and update maintenance-aiding materials to be used for interactive presentation to the technician. Prototype development and implementation plans are presented for two application areas: the Precision Measurement and Equipment Lab (PMEL) and F-16 checkout. Keywords: Job performance aiding, Electronic documentation, Authoring systems. (SDW)

Descriptors: *Maintenance management; *Supply depots; Electronics; Interfaces; Jobs; Laboratories; Measurement; Precision; Prototypes; Supervisors; Maintenance personnel

Identifiers: *Job aids; NTISDODXA

Section Headings: 70B (Administration and Management--Management Practice); 70D (Administration and Management--Personnel Management, Labor Relations, and Manpower Studies); 88B (Library and Information Sciences--Information Systems); 92A (Behavior and Society--Job Training and Career Development)

14/5/21 (Item 1 from file: 144)

DIALOG(R) File 144:Pascal

(c) 2004 INIST/CNRS. All rts. reserv.

13969480 PASCAL No.: 99-0151862

On-line computer system to minimize laser injuries during surgery : preliminary system layout and proposal of the key features

CANESTRI F

Hewlett-Packard Company GmbH, PO Box 14 30, 71004 Boeblingen, Germany

Journal: Proceedings of the institution of mechanical engineers.Part H.

Journal of engineering in medicine, 1999, 213 (1) 69-76

ISSN: 0954-4119 Availability: INIST-6044A8; 354000074458780070

No. of Refs.: 11 ref.

Document Type: P (Serial) ; A (Analytic)

Country of Publication: United Kingdom

Language: English

The aim of this paper is to investigate some new user interface ideas and related application packages which aim to improve the degree of safety in an operating room during surgical operations in which an invasive laser beam is deployed. The overall value of the proposition is that a means is provided which ensures the successful completion of the surgical case while minimizing the risk of thermal and mechanical injuries to healthy tissues adjacent to the surgical field. According to surgeons operating with a variety of CO SUB 2 lasers available at both the National Cancer Institute in Milan, Italy, and the Sackler School of Medicine, Tel Aviv University, Israel, each laser device presents different cutting and coagulation properties. In order to identify which 'ideal' procedure might corroborate the subjective impression of each surgeon and also to provide one common tool to ensure procedures with a high level of safety, the author has worked for several months with surgeons and technicians of both Institutions to define the general design of a new on-line surgical operation planning and design system to be used during the pre-operative briefing activities and also as a consultation tool during operation. This software package will be developed and tested on both 'C' and FORTRAN **compilers** running on a commercially available PC which is driving a continuous wave (CW) CO SUB 2 laser device via its Instrument Bus

interface. The present proposal describes the details of a software package called LCA (Laser-beam Controller and Adviser) which performs several controls in parallel on the key output parameters of a laser beam device during its utilization in delicate surgical operations. The required performances of this device needed during a given surgical operation are pre-simulated and compared against the well-known safety limits, which are stored in the computer's mass storage. If the surgeon's decision about the laser device set-up are considered to be too close to the required physiological safety limits, then the SW alerts the surgeon and proposes alternatives based on other combinations of both HW and SW configurations. An additional application of LCA uses this SW as a warning tool during the operation itself. If a wrong set-up of the laser device is accidentally used, an alarm will be generated and the laser beam automatically switched-off prior to an 'incident report' printout. The operation will continue only when the surgeon validates the choices that the SW suggests for implementation. If necessary, the surgeon can switch off the device and continue to operate it manually. In this case, the surveillance mode will be totally excluded.

English Descriptors: **Associated** technique; Laser; Surgery; On line device
; Computer aid; User **interface** ; Risk analysis; **Implementation** ;
Mechanical degradation; Thermal degradation
Broad Descriptors: Biomedical data processing; Informatique biomédicale;
Informatica biomedical

File 275:Gale Group Computer DB(TM) 1983-2004/Jun 25
(c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jun 25
(c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jun 24
(c) 2004 The Gale Group
File 16:Gale Group PROMT(R) 1990-2004/Jun 25
(c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
(c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2004/Jun 25
(c)2004 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Jun 24
(c) 2004 McGraw-Hill Co. Inc
File 15:ABI/Inform(R) 1971-2004/Jun 27
(c) 2004 ProQuest Info&Learning
File 647:CMP Computer Fulltext 1988-2004/Jun W2
(c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jun W2
(c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jun 28
(c) 2004 The Dialog Corp.
File 369:New Scientist 1994-2004/Jun W3
(c) 2004 Reed Business Information Ltd.

Set	Items	Description
S1	1498847	INTERFACE? ? OR DECLARATION? ?
S2	2727204	CLASS OR CLASSES OR IMPLEMENTATION? ?
S3	5462	S1(10N)S2(10N)(ASSOCIAT? OR CORRELAT? OR MAP???? OR MATCH?- ?? OR RELATE? ? OR RELATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNECT??? OR AFFILIAT? OR BIND???)
S4	289	(SIGNATURE OR NAME)()MATCHING OR CALLING()CONTEXT
S5	0	DEFAULT(3N)(ASSOCIATION()RULE? ?)
S6	0	OVERRIDE(3N)(ASSOCIATION()RULE? ?)
S7	332511	MSIL OR IL OR INTERMEDIATE()LANGUAGE OR BYTECODE
S8	116741	RUNTIME OR RUN()TIME OR CL OR CLS
S9	396817	COMPIL??????
S10	1135	ASSOCIATION()RULE? ?
S11	434	S3(100N)S4:S10
S12	1380	(INTERFACE? ? OR DECLARATION? ?)(7N)IMPLEMENTATION(7N)(ASS- OCIAT? OR CORRELAT? OR MAP???? OR MATCH??? OR RELATE? ? OR RE- LATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNEC- T??? OR AFFILIAT? OR BIND???)
S13	52	S12(50N)S4:S10
S14	35	RD (unique items)

14/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
© 2004 The Gale Group. All rts. reserv.

02404573 SUPPLIER NUMBER: 62503015 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Ikos takes emulation to RTL.(Product Announcement)
Santarini, Michael
Electronic Engineering Times, 61
June 5, 2000
DOCUMENT TYPE: Product Announcement ISSN: 0192-1541 LANGUAGE:
English RECORD TYPE: Fulltext
WORD COUNT: 620 LINE COUNT: 00052

... the ease of use of a software simulator," said Juergen Jaeger, director of product marketing. "But it goes beyond that by adding a co-modeling **interface** that allows us to directly **link** C architectural models running on a workstation directly to **implementation** details running on the emulator."

The system comprises the VLE-5M emulator and two new product elements: the Ikos RTL **Compiler** and the Transaction Interface Portal (TIP).

Jaeger said the RTL **Compiler** technology allows design teams to feed their RTL directly to the emulator. Previously, designers would have to use a logic synthesis tool, such as Synopsys...

14/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
© 2004 The Gale Group. All rts. reserv.

02323482 SUPPLIER NUMBER: 55506981 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Distributed objects; Roy Tynan works through a Java RMI server and client applet.(Technology Information)
EXE, 47(1)
August 2, 1999
ISSN: 0268-6872 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 2028 LINE COUNT: 00172

... implement the same interface, they can be used interchangeably by Java code. The RMI system works smoothly with Java programs because the stub and service **implementation** classes show the same **interface**. However, behind the **interface** they provide completely different implementations. The service **implementation** provides the behaviour while the stub file **implementation** provides a hidden **link** into the RMI system.

A TimeServer exampleWe are ready to build our TimeServer service. To do this, we first **compile** the interface TimerServer.java and the implementation, TimerServerImpl.java. Then we must run a utility program rmic to generate the stub and skeleton class files...

14/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
© 2004 The Gale Group. All rts. reserv.

0240333 SUPPLIER NUMBER: 54622881 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Making sense of the COM-CORBA debate.(Technology Information)
yee, andre
UNIX Review's Performance Computing, 17, 6, D3(1)
June, 1999
LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 2900 LINE COUNT: 00240

... Similarly, each interface corresponds to a GUID known as an interface ID (IID). Through the use of GUIDs, each object is referenced and invoked at **run time** by DCOM. The association between the GUID information and the host server is stored in the registry through a registration process. Every server that hosts DCOM objects must be

registered in order for those objects to be available to client components. With CORBA, a similar **association** is made between the **interface** name and the server, but this **association** is registered with an **implementation** repository instead. The ORB determines at **runtime** the physical location of the object by searching the implementation repository.

The differences between how DCOM and CORBA create objects are not significant from a...

14/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02202851 SUPPLIER NUMBER: 20913738 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Architecture of Posix and OpenNT. (Unix-to-Windows NT portability)
(includes a related article on using DataFocus's Nutcracker application development software) (Technology Information) (Cover Story)

Cannistra, William J.

UNIX Review's Performance Computing, v16, n6, p37(7)

June, 1998

DOCUMENT TYPE: Cover Story LANGUAGE: English RECORD TYPE:

Fulltext; Abstract

WORD COUNT: 3245 LINE COUNT: 00282

... DLL: directs the communications between the subsystem environment and OpenNT processes(6)

Processes and Signals

OpenNT is not an emulation of the UNIX environment. Applications **compiled** using the OpenNT SDK can make direct calls into the Windows NT kernel. For example, forks are supported using direct-kernel connectivity--bypassing the Win32 subsystem to spawn processes.

OpenNT has a POSIX.1 safe-signals **implementation**. It also provides the BSD and SVID access-signal **interfaces**, all **mapped** to the POSIX.1 **implementation**. The call to reset a signal handler (required practice in the traditional SVID model) is no longer essential. Therefore, source code should port directly with...

14/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02119768 SUPPLIER NUMBER: 19960202 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The COM+ programming model makes it easy to write components in any language. (Microsoft Component Object Model) (Technology Tutorial) (Tutorial)

Kirtland, Mary

Microsoft Systems Journal, v12, n12, p19(9)

Dec, 1997

DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: English

RECORD TYPE: Fulltext; Abstract

WORD COUNT: 6151 LINE COUNT: 00534

... inheritance.

In interface inheritance, a cointerface can inherit from one or more existing cointerfaces. You then implement the cointerface in a coclass. In the C++ **compiler** provided with the COM+ SDK, you use the keyword **implements** to indicate that a coclass implements one or more cointerfaces. If you want to create objects of that class, you must provide implementations of all methods and properties defined by the cointerfaces and any cointerfaces they inherit.

Implementation inheritance is a little different. In this **relationship**, a coclass inherits both **interface** and **implementation** from another coclass. COM+ will let you use single **implementation** inheritance within a process. COM+ does not permit multiple inheritance of coclasses. Furthermore, COM+ classes may only inherit from other COM+ classes, and non-COM...

14/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02109753 SUPPLIER NUMBER: 19802407 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Q&A: ActiveX/COM. (10 myths about COM) (Technology
Tutorial) (Column) (Technical)
Box, Don
Microsoft Systems Journal, v12, n11, p73(6)
Nov, 1997
DOCUMENT TYPE: Column Technical ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 4431 LINE COUNT: 00385

... one for the call to Advise). Because the Advise method implied
above would marshal the correct type of interface based on the IID
specified at **runtime**, no additional round-trips would be required in the
object's **implementation** of Advise.

Another limitation of the **connection point interfaces** (which also
applies to the IConnectableObject **interface** in the previous code) is that
the Advise/Unadvise methods are very generic. It is impossible to provide
additional parameters to the connection request to...

14/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02040948 SUPPLIER NUMBER: 19168006 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PC size, mainframe power. (trends in distributed objects and naming
services) (Looking Forward: Technology on the Way) (Company Business and
Marketing) (Cover Story)
Seitzer, Larry
PC Magazine, v16, n6, p200(3)
March 25, 1997
DOCUMENT TYPE: Cover Story ISSN: 0888-8507 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 2203 LINE COUNT: 00180

... including IBM's SOMobjects (one is coming from Netscape). Most of
CORBA's architecture is like DCOM's, but CORBA adds a new layer: the
Interface Definition Language (IDL). The IDL defines the **interface** of
the object separate from its **implementation**. Developers **compile** the IDL
into **interface** code, which **connects** to the object broker and
implementation code on both systems. This code can be written in almost
any language, and the data type mappings between the implementation code
and IDL are...

14/3,K/8 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01906956 SUPPLIER NUMBER: 18043452 (USE FORMAT 7 OR 9 FOR FULL TEXT)
SUNSOFT PREPARES JAVA OBJECT REQUEST BROKER WITH ABILITY TO DOWNLOAD
INTERACTIVE JAVA APPLETS.
Computergram International, n862, pCGN02290009
Feb 19, 1996
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 446 LINE COUNT: 00040

TEXT:

...redeployed as a Web server application's requirements evolve, the
firm says. The server application can be written in any language for which
a Corba **Interface Definition Language** language **mapping** exists. Client
applications are written without regard for the server **implementation**.
SunSoft is also preparing versions of its Workshop development tools

tailored for Java. The licensing terms for the Java Request Broker and the Interface Definition Language **compiler** will be similar to those for Java itself.

14/3,K/9 (Item 9 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01805386 SUPPLIER NUMBER: 17180696 (USE FORMAT 7 OR 9 FOR FULL TEXT)
OLE. (Q&A) (Tutorial)
Box, Don
Microsoft Systems Journal, v10, n8, p77(10)
August, 1995
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 3152 LINE COUNT: 00340

... succession. Because such a loop would literally take decades to complete, even on an in-proc object, some other approach is in order.

The best **implementation** would test only the IIDs that actually **correspond** to real **interfaces**. The IIDs for the standard interfaces are listed in the system header files, but the IIDs for custom interfaces are not, so statically building the list at **compile** time won't work. One way to dynamically find all bound IIDs is to look at a section of the registry that lists the Proxy...

14/3,K/10 (Item 10 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01718069 SUPPLIER NUMBER: 16367489
OLE's fragmented inheritance. (object-oriented programming) (the Meta View) (Technical) (Column)
Babcock, Charles
Computerworld, v29, n3, p8(1)
Jan 16, 1995
DOCUMENT TYPE: Column ISSN: 0010-4841 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

...ABSTRACT: Object Model (COM) both do not have a descriptive language like C++ that draws things together, but similar results are achieved by dividing the object **interfaces** from the object **implementation**. OLE objects have pointers added to the unknown **interface** that is common to all OLE objects instead of **links** being made under the covers at **compile** time.

14/3,K/11 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01545349 SUPPLIER NUMBER: 12780458 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Questions & answers: C/C++. (Tutorial)
Sinofsky, Steven
Microsoft Systems Journal, v7, n8, p85(4)
Dec, 1992
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2010 LINE COUNT: 00149

... DLL mechanism in mind. Mixing the two is, at best, difficult.

There are two things to consider. First is the mechanics of writing a C++ **interface** DLL, which is pretty straightforward. Second is the design and **implementation** associated with a C++ DLL.

For the first part, both C/C++ 7.0 and BC++ 3.1 have essentially the same implementation and **compiler** feature set for exporting class

interfaces. You declare a class with the [underscore]export keyword, which goes after the class keyword; all of its member...

14/3,K/12 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01520550 SUPPLIER NUMBER: 12324055 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Feedback. (Letter to the Editor)
Parker, H. Dan; Throop, Wayne; Johns, Vincent R.
Computer Language, v9, n7, p11(5)
July, 1992
DOCUMENT TYPE: Letter to the Editor ISSN: 0749-2839 LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1741 LINE COUNT: 00136

... not cast the pointer values, despite carefully setting up prototypes that might be thought to do so. Even though a prototype is in scope, the **interface** prototype does not **match** the **implementation** prototype (because the HSET type is defined differently in clients vs. implementors). In fact, to be blunt, the code is simply lying to the **compiler** about what the client is going to do.

Using Abraxis' CodeCheck, we find:

Checking file "Sets.c" with rules from "ansi.cc"... # File "Sets.c

...

14/3,K/13 (Item 13 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01465389 SUPPLIER NUMBER: 11622346 (USE FORMAT 7 OR 9 FOR FULL TEXT)
TopSpeed V3.0 - C++ and Pascal. (program development software from JPI)
(Software Review) (includes related article on the Rouge Wave class library) (Evaluation)
Smith, Paul G.
EXE, v6, n5, p70(4)
Oct, 1991
DOCUMENT TYPE: Evaluation ISSN: 0268-6872 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3103 LINE COUNT: 00246

... actual bodies of the procedures and functions defined in the interface.

The syntax of TopSpeed Pascal units is, therefore, a little different to other Pascal **compilers**. An example **interface** unit is shown in Figure 1. Its **corresponding implementation** unit would commence with the heading '**IMPLEMENTATION UNIT COMMS**' and continue, after defining any private constants, types, procedures, and functions, by defining the bodies of the procedures and functions defined in the...

14/3,K/14 (Item 14 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01452940 SUPPLIER NUMBER: 11315493 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Paradox and dBase reborn. (Borland International Inc. acquires Ashton-Tate)
(Forum) (Brief Article)
Gliedman, John
PC Sources, v2, n10, p53(1)
Oct, 1991
DOCUMENT TYPE: Brief Article ISSN: 1052-6579 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 242 LINE COUNT: 00019

... compatible with their DOS counterparts. NO release dates are set.

In addition, Borland announced that it is committed to providing users with a dBase language **compiler** and interpreter for DOS, Windows, and future environments, and will continue to upgrade the current DOS product line.

WYSIWYG **interfaces**, full SQL **implementation** (Borland already provides this for Paradox with its SWL **Link** add-on), and object-oriented application creation should give pause to all competitors, especially Fox Software.

14/3,K/15 (Item 15 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01442638 SUPPLIER NUMBER: 11037697 (USE FORMAT 7 OR 9 FOR FULL TEXT)
CDM Windows V3.2. (Database Technologies) (New Products) (product announcement)

C Users Journal, v9, n8, p122(1)

August, 1991

DOCUMENT TYPE: product announcement ISSN: 0898-9788 LANGUAGE:

ENGLISH RECORD TYPE: FULLTEXT

WORD COUNT: 165 LINE COUNT: 00013

... level part of this product is based on Btree algorithms to guarantee fast sequential and random data access. Features include . dynamic data definition, rich entity **relationship** model, network database model, flexible and easy-to-use high-level programming **interface** and **implementation** of complex queries. CDM also supports variable length records and multiple keys.

Runtime distribution is royalty-free. Complete source code is \$895. For more information, contact Database Technologies, 213 Reservoir Rd, Brookline, MA 2167 (617) 739-3390.

14/3,K/16 (Item 16 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01414156 SUPPLIER NUMBER: 09723103 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Uniform preview: Open Inc pitches into graphical user interface fray with Aspect.

Computergram International, n1586, pCGI01090009

Jan 9, 1991

ISSN: 0268-716X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT

WORD COUNT: 985 LINE COUNT: 00079

TEXT:

...the Mac graphical user interfaces - as well as on character-based terminals. Applications written to the Aspect application programming interface can be tailored for a **run - time** environment for each of the interfaces. For example, an application with an Aspect user **interface** when **linked** with the Aspect Motif **run - time** produces a native Motif **implementation**. Aspect's design tool stores specifications for the **interface** in a database. These specifications are not bound into the application, but are retrieved by the Aspect **run - time** system - a configuration that Open claims requires less of the interface to be hard-coded and allows changes to be made to the database without the need to re-**compile** applications. Aspect also enables the same applications to be run on character-based terminals as on systems with graphical user interfaces. This is because a...

14/3,K/17 (Item 17 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01289563 SUPPLIER NUMBER: 07125676 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Ada out of uniform. (Software Review) (Ada compilers) (evaluation)

Dortenzo, Megan

PC Tech Journal, v7, n4, p86(15)

April, 1989

DOCUMENT TYPE: evaluation ISSN: 0738-0194

LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 8221 LINE COUNT: 00673

... is very powerful and is recommended for any developers who can justify its price and system requirements. An application is only as good as its **compiler** : large, complex Ada applications need the capabilities that Alsys provides. MERIDIAN ADAVANTAGE Developers can access the versatile AdaVantage environment from DOS, similar to the access of mainframe Ada systems, or through the completely contained environment called Ada Developer's **Interface** (ADI). Also similar to a mainframe **implementation** , you issue library-management, **compile** , and **link** commands independently of each other in succession at the DOS prompt. A programmer familiar with DEC's Ada **Compilation** System will have no difficulty with AdaVantage. As an integrated environment, AdaVantage provides additional support tools.

The AdaVantage PC Developer Kit consists of the **compiler** , the ADI, linker, library manager, symbolic debugger, and the predefined packages DOS and Utilities, and costs \$1,095. The AdaVantage PC Professional Developer Kit includes...

14/3,K/18 (Item 18 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01183419 SUPPLIER NUMBER: 06091248

Hewlett-Packard Precision Architecture compiler performance. (technical)

Pettis, Carl W.; Buzbee, William B.

Hewlett-Packard Journal, v38, n3, p29(7)

March, 1987

DOCUMENT TYPE: technical

ISSN: 0018-1153

LANGUAGE: ENGLISH

RECORD TYPE: ABSTRACT

ABSTRACT: The new HP Precision Architecture provides a low-level interface to hardware through a fast instruction set. **Compilers** must be developed for this new architecture to provide high-level language **interfaces** to the machine. The influence of the performance criteria **related** to the **implementation** of the new **compilers** is discussed, first that of high-level languages on the design of the instruction set, and second that of the handling of truly complex operations...

14/3,K/19 (Item 1 from file: 621)

DIALOG(R)File 621:Gale Group New Prod. Annou. (R)

(c) 2004 The Gale Group. All rts. reserv.

03150071 Supplier Number: 84137334 (USE FORMAT 7 FOR FULLTEXT)

IONA Announces Orbix E2A XMLBus Edition v5.1 - Extending Its Leadership in Web Services Innovation for Real-World Business.

Business Wire, p2203

March 25, 2002

Language: English Record Type: Fulltext

Document Type: Newswire; Trade

Word Count: 879

... and deploy Web services that represent multiple objects and

components. Using the Designer, users can easily combine a series of calls to

multiple fine grained **implementation** objects into a single Web Service. At

runtime , The **Interface** Transformation Engine automatically **maps** the

implementation types to a well-formed Web service **interface** .

- Business Process Engine provides a visual environment for coordinating business tasks into a single executable process flow and exposing it as a Web service. In...

14/3,K/20 (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01768528 Supplier Number: 53354825 (USE FORMAT 7 FOR FULLTEXT)
Wind River Systems Provides Comprehensive Digital Imaging Solution to Minolta Corporation.
PR Newswire, p2133
Dec 7, 1998
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 914

... writing application code for specific features. All the Tornado tools share a common, graphical look and feel. In addition, a rich set of application programming **interfaces** (APIs) ranging from the graphical user **interface** to the host-target **connection implementation** is published and available for reference. These published APIs enable Minolta's software developers to tailor Tornado for specific digital imaging applications by easily integrating custom or third-party tools as well as **runtime** components such as Adobe's PostScript, innovative connection protocols like Hewlett Packard's JetSend, and camera technology such as FlashPoint's Digita operating environment.
"Reliability..."

14/3,K/21 (Item 3 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01512556 Supplier Number: 47266558 (USE FORMAT 7 FOR FULLTEXT)
Mitsubishi endorses Sun's EmbeddedJava API and PersonalJava API as critical platforms for Mitsubishi's "Java in eRAM" strategy.
Business Wire, p04020158
April 2, 1997
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 654

... a set of class libraries that require approximately 512 Kbytes of memory space. EmbeddedJava also includes optional Java modules that can be individually "mixed and **matched** " to meet end-application requirements.

PersonalJava is an application programming **interface** (API) and **implementation** of Java for networked products, such as Web-browsing phones, Web televisions PDAs, office automation equipment and consumer Information Appliances that do not require the complete set of enterprise connectivity supported by the full Java **Runtime Environment**.

PersonalJava is designed for network-centric devices with varying display types, including television and TFT LCDs, and with input and navigation devices other than...

14/3,K/22 (Item 4 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01157171 Supplier Number: 41920041 (USE FORMAT 7 FOR FULLTEXT)
THE EXTOL EDI INTEGRATOR, COMPREHENSIVE EDI MANAGEMENT SOFTWARE FOR THE IBM AS/400, IS NOW AVAILABLE FROM EXTOL, INC.
News Release, p1

March 8, 1991

Language: English Record Type: Fulltext

Document Type: Magazine/Journal; Trade

Word Count: 672

... the user's application database with the EDI system.

The Extol EDI Integrator provides tremendous flexibility in user file definition, code translation options and data **mapping**, including Application Program **Interface** (API) points for easy customization.

As EDI **implementation**

expands within an organization, data management issues and the speed at which transactions can be processed become increasingly important. The Extol EDI Integrator includes a comprehensive data management system including audit trail data structures which are automatically maintained on all primary system files. The translator executes utilizing a **compiled** state table, an advanced design which provides high speed translation performance.

The code translation capabilities of the system include the ability to access existing code...

14/3,K/23 (Item 1 from file: 636)

DIALOG(R)File 636:Gale Group Newsletter DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

04858182 Supplier Number: 67689999 (USE FORMAT 7 FOR FULLTEXT)

Legal Counsel briefs Security Council on efforts to combat terrorism;

Reviews international legal regime, stresses need for cooperation.

M2 Presswire, pNA

Dec 7, 2000

Language: English Record Type: Fulltext

Document Type: Newswire; Trade

Word Count: 3904

... extensive provisions on international cooperation.

Focusing on the role of the Secretariat, he said the principle means by which the Secretary-General assisted in the **implementation** of the 1994

Declaration had been an annual report containing data on the status and **implementation** of existing multilateral, regional and bilateral arrangements **relating** to international terrorism. There are currently 19 global or regional treaties pertaining to international terrorism. The Secretary-General also **compiles** a compendium of national laws and regulations regarding the prevention and suppression of international terrorism. The Office of Legal Affairs also **compiles** global and regional conventions on international terrorism. An

14/3,K/24 (Item 2 from file: 636)

DIALOG(R)File 636:Gale Group Newsletter DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

02552883 Supplier Number: 45150528 (USE FORMAT 7 FOR FULLTEXT)

ACE UNVEILS REVOLUTIONARY COMPILER TECHNOLOGY FOR SPARC; AUTOMATIC CODE

GENERATION SPEEDS PRODUCTION

Computergram International, n2548, pN/A

Nov 21, 1994

Language: English Record Type: Fulltext

Document Type: Newswire; Trade

Word Count: 226

(USE FORMAT 7 FOR FULLTEXT)

TEXT:

...with the new technology, it has reduced production time by a factor of four. The product, which is based on the user -transparent CoSy parallel **Compilation** System, is available now on CD-ROM for the Solaris 2

implementation of Unix. The Expert Sparc **Compilers** **interface** to the

Solaris **linkage** editor and to popular debuggers such as TotalView and dbx. The compact disk contains all five Expert Sparc **Compilers**, plus all the necessary associated documentation and will sell for \$500. The company is now working on similar CoSy **compilers** for the Intel Corp Pentium and the PowerPC RISC. Other processors are targeted for next year, says the company.

14/3,K/25 (Item 3 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02094458 Supplier Number: 43868553 (USE FORMAT 7 FOR FULLTEXT)
OS/2: WOLLONGONG SHIPS PATHWAY ACCESS FOR OS/2
EDGE: Work-Group Computing Report, v4, n159, pN/A
May 31, 1993
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 535

... Ring networks. The menu-driven installation and configuration utility simplifies the setup and customization process. PathWay API 1.2 provides a 32-bit Windows dynamic **link** library (DLL) **implementation** of Berkeley Sockets. It includes an **interface** for IBM C Set/2 and support for out-of-band or urgent data.

Wollongong offers PathWay Access 2.1, which comes bundled with PathWay **Runtime**, for \$350 per single license. PathWay **Runtime** 1.2 is \$195 per single license. Discount prices are available for volume orders. PathWay API 1.2 is \$200. For more information, call Wollongong...

14/3,K/26 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

06673699 Supplier Number: 55871072 (USE FORMAT 7 FOR FULLTEXT)
Services : Ground control networks.
Geo Info Systems, v9, n8, p36
August, 1999
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 63

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
Aerial Data Service offers ground control networks and digital photogrammetric base **mapping** for GIS **implementation**. The company's **map compilers** collect digital planimetric, elevation model, and terrain model data. Stereoplotters **interface** with interactive graphic systems for map digitizing; superimposition creates a photographic model of the ground and map data. Each project is designed according to client...

14/3,K/27 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

04876509 Supplier Number: 47173666 (USE FORMAT 7 FOR FULLTEXT)
Into Orbit: Object Request Brokers: Servers Of The 21st Century
Frey, Anthony
Network Computing, p51
March 1, 1997
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 3571

... itself typically needs to be installed on every host. Although the products we tested were full-development versions that had the Interface

Definition Language (IDL) **compiler** and associated header files necessary for language **mapping**, you'll probably need to install only the ORB, the **Interface** Repository and the **Implementation** Repository that are included in **run - time** versions.

We encountered no disturbances during the ORB installation process, which is really quite simple, since most ORBs are implemented as one or two individual...

14/3,K/28 (Item 1 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)
(c) 1999 The Gale Group. All rts. reserv.

01193027

Modula-2 aids structured programming.
MINI MICRO SYSTEMS May, 1985 p. 77-82

... by allowing several programmers to work simultaneously on different parts of a program. It has a more streamlined English syntax than does Pascal and uses **compilation** units called definition and **implementation** modules to solve problems **associated** with partitioning. Definition modules define the **interface** between a module and the external world and contain declarations of objects that are essential to their definition. Implementation modules implement the definition module's...

14/3,K/29 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

10166241 SUPPLIER NUMBER: 20217862 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Objects push a paradigm shift in distributed computing.
Gien, Michel
Electronic Design, v45, n23, p125(6)
Oct 23, 1997
ISSN: 0013-4872 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 3929 LINE COUNT: 00329

... code are generated according to a programming language mapping for IDL, in our case the C++ mapping. How this is done depends on the IDL **compiler**. For the purpose of this example we will assume that the following four files are generated:

console. H - header file for the stub for the Console **interface**
console. C - **implementation** of the stub, **compiled** and **linked**
with both client and server programs
sk...

14/3,K/30 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

9710889 SUPPLIER NUMBER: 19727740 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Wind River Systems Enters \$3.4 Billion Programmable DSP Market
PR Newswire, p908SFM030
Sep 8, 1997
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1493 LINE COUNT: 00135

... environment that includes a full range of Wind River and third party development tools. To facilitate integration of non-Wind River tools, several application programming **interfaces** (APIs) are available and published for reference, from GUI **interfaces** to **connection implementation**.

The Tornado for DSP tool suite offers a GNU-based **compiler**, **linker** and assembler; the CrossWind(TM) debugger for task-aware and system level debugging; a browser for graphically displaying WiSP code, objects, memory allocation, message queues...

14/3,K/31 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

01847827 04-98818

The Data Protection Bill 1998: A comparative examination

Slee, David

Information & Communications Technology Law v8n1 PP: 71-109 Mar 1999

ISSN: 1360-0834 JRNL CODE: ICTL

WORD COUNT: 10634

...TEXT: inevitable. Incorporation by statute such as the current Bill, as noted above, at least allows the government a degree of control over the manner of **implementation**. For example cl .3 the **relationship** of the convention to existing legislation; cl .4 the '**Declaration** of Incompatibility.'

6. Council of Europe Convention on the Protection of Individuals with regard to the Automatic Processing of Personal Data 108/81 28 January...

14/3,K/32 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

01706532 03-57522

Evaluating architectures for multithreaded object request brokers

Schmidt, Douglas C

Communications of the ACM v41n10 PP: 54-60 Oct 1998

ISSN: 0001-0782 JRNL CODE: ACM

WORD COUNT: 3873

...TEXT: about IDL interfaces. Using this information, it is possible for a program to encounter an object whose interface was not known when the program was **compiled**, yet be able to determine what operations are valid on the object and make invocations on it. In addition, the Interface Repository provides a common location to store additional information **associated** with **interfaces** ' ORB objects, such as stub/skeleton type libraries.

The **Implementation** Repository contains information that allows the ORB to locate and activate servants. Most of the information in the Implementation Repository is specific to an ORB...

14/3,K/33 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2004 CMP Media, LLC. All rts. reserv.

01216574 CMP ACCESSION NUMBER: EET20000605S0034

Ikos takes emulation to RTL

Michael Santarini

ELECTRONIC ENGINEERING TIMES, 2000, n 1116, PG61

PUBLICATION DATE: 000605

JOURNAL CODE: EET LANGUAGE: English

RECORD TYPE: Fulltext

SECTION HEADING: Design Automation

WORD COUNT: 576

... the ease of use of a software simulator," said Juergen Jaeger, director of product marketing. "But it goes beyond that by adding a co-modeling **interface** that allows us to directly **link** C architectural models running on a workstation directly to **implementation** details running on the emulator."

The system comprises the VLE-5M emulator and two new product

elements: the Ikos RTL **Compiler** and the Transaction Interface Portal (TIP).

Jaeger said the RTL **Compiler** technology allows design teams to feed their RTL directly to the emulator. Previously, designers would have to use a logic synthesis tool, such as Synopsys...

14/3,K/34 (Item 2 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2004 CMP Media, LLC. All rts. reserv.

01092791 CMP ACCESSION NUMBER: WIN19960701S0148
Windows and the Web Are Converging - Microsoft is finding ways to survive
and thrive in an Internet-dominated world. (Programming Windows)
Martin Heller
WINDOWS MAGAZINE, 1996, n 707, PG265
PUBLICATION DATE: 960701
JOURNAL CODE: WIN LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: How To
WORD COUNT: 1656

... controls, learn about dual interfaces. As I mentioned earlier, an OLE Automation object can expose an IDispatch interface, which is a general form of late (**runtime**) binding. OLE Automation objects can also expose a VTBL interface, which is a more efficient early (**compile-time**) binding mechanism. Dual **interface** OLE Automation objects expose both IDispatch and VTBL **interfaces**. The general dual **interface** **implementation** scheme for us lazy programmers is to delegate as much as possible. If you're writing an MFC automation object with dual interfaces, you can...

14/3,K/35 (Item 3 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2004 CMP Media, LLC. All rts. reserv.

00551356 CMP ACCESSION NUMBER: OST19930215S9739
HP Tool Showcases Key Object Spec-Product Enhancements From DEC Could Also
Add Credibility To Object Request Broker Specification
DAN RICHMAN
OPEN SYSTEMS TODAY, 1993, n 117, 5
PUBLICATION DATE: 930215
JOURNAL CODE: OST LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: news
WORD COUNT: 916

... 5,000. For OpenVMS AXP, development licenses will list for between \$4,000 and \$28,000, and for System 7 they'll list for \$300. **Run-time** licenses (one is required on each machine) will list for \$300 for HP-UX and AIX, \$100 for Mac and from \$300 to \$1,200 for OpenVMS AXP, depending on machine size.

ACA Services 2.1 is a less complete **implementation** of CORBA 1.1 than HP's, lacking static **binding** and the standard **interface** definition language. It includes a language called CRL (Class Repository Language), which DEC said offers capabilities similar to IDL. DEC's ORB can work with...

File 348:EUROPEAN PATENTS 1978-2004/Jun W03

(c) 2004 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20040624,UT=20040617

(c) 2004 WIPO/Univentio

Set	Items	Description
S1	264752	INTERFACE? ? OR DECLARATION? ?
S2	273084	CLASS OR CLASSES OR IMPLEMENTATION? ?
S3	3991	S1(10N)S2(10N)(ASSOCIAT? OR CORRELAT? OR MAP???? OR MATCH?- ?? OR RELATE? ? OR RELATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNECT??? OR AFFILIAT? OR BIND???)
S4	250	(SIGNATURE OR NAME)()MATCHING OR CALLING()CONTEXT
S5	0	DEFAULT(3N)(ASSOCIATION()RULE? ?)
S6	0	OVERRIDE(3N)(ASSOCIATION()RULE? ?)
S7	462419	MSIL OR IL OR INTERMEDIATE()LANGUAGE OR BYTECODE
S8	193719	RUNTIME OR RUN()TIME OR CL OR CLS
S9	29205	COMPIL??????
S10	145	ASSOCIATION()RULE? ?
S11	328	S3(50N)S4:S10
S12	286	S11 AND IC=G06F
S13	104	DECLARATION? ?(7N)S2(7N)(ASSOCIAT? OR CORRELAT? OR MAP???? OR MATCH??? OR RELATE? ? OR RELATING OR RELATIONSHIP? ? OR CO- RRESPOND? OR LINK??? OR CONNECT??? OR AFFILIAT? OR BIND???)
S14	21	S13(50N)S4:S10
S15	1	S3(100N)(S4 OR S10)(100N)S7:S9
S16	34	S3(30N)(S4 OR RULE? ? OR CRITERIA OR CRITERION)(30N)S7:S9
S17	30	S16 NOT S14
S18	25	S17 AND IC=G06F

14/3,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

01299586

SYSTEM AND METHOD SUPPORTING TYPE CHECKING OF OPTIONS
SYSTEM UND VERFAHREN ZUR UNTERSTUTZUNG VON OPTIONENTYPENKONTROLLE
SYSTEME ET PROCEDE DE CONTROLE DU TYPE D'OPTIONS

PATENT ASSIGNEE:

Curl Corporation, (3117690), 8th floor, 400 Technology Square, Cambridge,
MA 02139-3539, (US), (Proprietor designated states: all)

INVENTOR:

HALSTEAD, Robert, H., Jr., 24 Louise Road, Belmont, MA 02478, (US)
KRANZ, David, A., 115 High Haith Road, Arlington, MA 02476, (US)
TERMAN, Christopher, J., 60 Cedar Street, Newton Center, MA 02459, (US)
WARD, Stephen, A., 199 Coolidge Avenue 803, Watertown, MA 02472, (US)

LEGAL REPRESENTATIVE:

Driver, Virginia Rozanne et al (58902), Page White & Farrer 54 Doughty
Street, London WC1N 2LS, (GB)

PATENT (CC, No, Kind, Date): EP 1226496 A2 020731 (Basic)
EP 1226496 B1 040114
WO 2001033346 010510

APPLICATION (CC, No, Date): EP 2000976698 001031; WO 2000US29853 001031

PRIORITY (CC, No, Date): US 162825 P 991101; US 672848 000928

DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;
LU; MC; NL; PT

EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI

INTERNATIONAL PATENT CLASS: G06F-009/44

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200403	1119
CLAIMS B	(German)	200403	970
CLAIMS B	(French)	200403	1290
SPEC B	(English)	200403	11438
Total word count - document A			0
Total word count - document B			14817
Total word count - documents A + B			14817

...SPECIFICATION c.remove-option <x>)

(is-set? c.x) becomes (c.option-set? <x>)

The notation <x> represents an "option binding" generated by the **compiler** that serves as a key for option storage and lookup. Each **class** that contains an option **declaration** for x has an **associated** option **binding** for x. When the **compiler** **compiles** a reference such as c.x, where c is an instance of a class C, the **compiler** locates the most specific option binding for x that is in the class C or one of its superclasses. The lookup algorithm used for this process is exactly the same as is used when **compiling** references to fields. Thus, if C itself contains a declaration of the option x, the **binding associated** with that **declaration** is used. If not, then if C's base **class** D has such a **declaration**, D's **binding** is used. If not, the algorithm proceeds in the same fashion until a declaration of x is located. The term "most specific option binding" is...

...that is reached by following the superclass-option-binding links repeatedly until the last OptionBinding in the chain is reached

next: used to build a **linked** list of the OptionBindings that **correspond** to **declarations** in a particular **class**

The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindings including bindings 30 and 32. Option binding...is a superclass

of Visual, which is in turn a superclass of every graphical class. Ignoring other classes not relevant to this discussion, the resulting **class** hierarchy looks like Figure 8.

If, say, the definition of the Rectangle **class** contains a **declaration** such as

```
(nonlocal-option background:any="white")
```

an option **binding** for background will also be created in the class GraphicOptions, which means that the background option will be able to be set not just on a Rectangle but also on a VBox, HBox, or any other Graphic.

Consider the following code whose **compilation** produces the data structure of Figure 9: We can then define the following subclasses: The above definitions can be seen to compile directly to the...

14/3,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00999921

METHOD AND APPARATUS FOR UPDATING AND SYNCHRONIZING INFORMATION BETWEEN A CLIENT AND A SERVER

VORRICHTUNG UND VERFAHREN ZUR AKTUALISIERUNG UND ZUR SYNCHRONISIERUNG VON INFORMATIONEN ZWISCHEN EINEM KLIENT UND EINEM SERVER

PROCEDE PERMETTANT UNE MISE A JOUR ET UN SYNCHRONISATION DE L'INFORMATION ENTRE UN UTILISATEUR ET UN SERVEUR ET DISPOSITIF CORRESPONDANT

PATENT ASSIGNEE:

APPLE COMPUTER, INC., (1211955), M/S 38-PAT, 1 Infinite Loop, Cupertino, CA 95014, (US), (Proprietor designated states: all)

INVENTOR:

FERRIS, Michael, c/o Apple Computer Inc. 1 Infinite Loop, M/S:38-PAT, Cupertino, CA 95014, (US)

POPP, Nicolas, 196 Santa Monica Avenue, Menlo Park, CA 94025, (US)

FORSTALL, Scott, 2349 Greer Road, Palo Alto, CA 94303, (US)

D'HARCOURT, Charles, 571 Haight Street, San Francisco, CA 94117, (US)

LEGAL REPRESENTATIVE:

Goddard, Heinz J., Dr. et al (4231), FORRESTER & BOEHMERT

Franz-Joseph-Strasse 38, 80801 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 972386 A1 000119 (Basic)

EP 972386 B1 010926

WO 9844695 981008

APPLICATION (CC, No, Date): EP 98912123 980330; WO 98US6227 980330

PRIORITY (CC, No, Date): US 825538 970331

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS: G06F-009/44; G06F-009/46; H04L-029/06

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
----------------	----------	--------	------------

CLAIMS B	(English)	200139	1853
----------	-----------	--------	------

CLAIMS B	(German)	200139	1925
----------	----------	--------	------

CLAIMS B	(French)	200139	2110
----------	----------	--------	------

SPEC B	(English)	200139	7774
--------	-----------	--------	------

Total word count - document A	0
-------------------------------	---

Total word count - document B	13662
-------------------------------	-------

Total word count - documents A + B	13662
------------------------------------	-------

...SPECIFICATION the object's class, for example.

The WEBOBJECT tags in Table One (INPUTFIELD, BUTTON, and OUTPUTFIELD) refer (or provide a pointer) to entries in a **declarations** file that **map** the entries to object **classes** and provide **declarations** for the tag. These **declarations** can be used to initialize instance variables of an object and provide **runtime** environment information such as object, method and variable bindings. Table Two provides an example of the contents of a declarations file that accompanies the HTML...

14/3,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00948568

Inter-object communication method
Interobjekt-Kommunikationsverfahren
Methode de communication interobject

PATENT ASSIGNEE:

Hitachi, Ltd., (204141), 6, Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo
101, (JP), (applicant designated states: DE;FR;GB)

INVENTOR:

Aoshima, Tatsundo, 17-12-A203, Yutakacho, Sagamihara-shi, (JP)
Hashimoto, Tetsuya, 11-10, Asagayakita-2-chome, Suginami-ku, Tokyo, (JP)

LEGAL REPRESENTATIVE:

Strehl Schubel-Hopf & Partner (100941), Maximilianstrasse 54, 80538
Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 860776 A1 980826 (Basic)

APPLICATION (CC, No, Date): EP 98102297 980210;

PRIORITY (CC, No, Date): JP 9751090 970219

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS: G06F-009/46

ABSTRACT WORD COUNT: 142

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9835	750
SPEC A	(English)	9835	5517
Total word count - document A			6267
Total word count - document B			0
Total word count - documents A + B			6267

...SPECIFICATION routing program 1r is described in source level language.
Even in this example, C++ is used as an example of the programming
language after the **compilation**.

The routing program 1r is generated by **compiling** a routing program
header file 1rfl and a routing program source file 1rf2.

The routing program header file 1rfl has an interface name variable
declaration (**corresponding** to "static.....InterfaceName" given below
the Routing Program **Class Declaration** , a queuing option variable
declaration 1rfl1 **compiled** from the queueing option description 4100
(see Fig. 6), and a **matching** method declaration 1rfl2 to be called by a
router 12 (see Fig. 1).

The routing program source file 1rf2 has an interface name variable
declaration (**corresponding** to "static....."DigitalLibrary"" given
below the Routing Program **Class Definition**), an external method
declaration 1rf20 **compiled** from the native method description 40, a
queuing option variable definition 1rf21 and a matching method definition
1rf22.

Shown in Fig. 5 is a flowchart...

14/3,K/6 (Item 6 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00879292

Incremental compilation of c++ programs
Inkrementelle Komprimierung von C++ Programmen
Compilation incrementale de programmes C++

PATENT ASSIGNEE:

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: DE;GB)

INVENTOR:

Nackman, Lee Richard, 1 Reynal Road, White Plains, N.Y. 10605, (US)

Karasick, Michael, 11-4 Brooke Club Drive, Ossining, N.Y. 10562, (US)
Barton, John Joseph, 13 Tennis Court Road, Mahopac, N.Y., (US)
Lieber, Derek, 2497 Welsch Court, Yorktown Heights, N.Y. 10598, (US)
Streeter, David Joseph, 97 Campania Crescent, Scarborough, Ontario, M1V
2E8, (CA)

LEGAL REPRESENTATIVE:

Burt, Roger James, Dr. (52152), IBM United Kingdom Limited Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 805391 A2 971105 (Basic)
APPLICATION (CC, No, Date): EP 97302677 970418;
PRIORITY (CC, No, Date): CA 2175711 960501
DESIGNATED STATES: DE; GB
INTERNATIONAL PATENT CLASS: G06F-009/45;
ABSTRACT WORD COUNT: 81

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9710W5	327
SPEC A	(English)	9710W5	8092
Total word count - document A			8419
Total word count - document B			0
Total word count - documents A + B			8419

...SPECIFICATION and each of these declarations might be parsed more than
once. When all of the definitions are not available in source form, for
example, in **class** libraries supplied in **compiled** form, the forward
declarations supplied in the **corresponding** header files can be used
instead.

According to the invention, the top-level declarations are defined to
be top-level declaration statements (e.g., global...

14/3,K/7 (Item 7 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00614518

SYSTEM FOR CHANGING SOFTWARE DURING COMPUTER OPERATION
SYSTEM ZUM ANDERN VON SOFTWARE WAHREND DES RECHNERBETRIEBS.
SYSTEME PERMETTANT LE REMPLACEMENT D'UN LOGICIEL PENDANT LE FONCTIONNEMENT
D'UN ORDINATEUR

PATENT ASSIGNEE:

TELEFONAKTIEBOLAGET LM ERICSSON, (213761), , 126 25 Stockholm, (SE),
(Proprietor designated states: all)

INVENTOR:

NILSSON, Kjell, Rickard, Katarina Bangatan 71, 2trp, S-116 39 Stockholm,
(SE)

MARKSTROM, Ulf, Kurt, Haakan, Vanadisvagen 32, 3trp, S-113 46 Stockholm,
(SE)

KLOFVER, Jan, Leif, Ingemar, Baagspannarvagen 5, S-125 30 Alvsjo, (SE)

LEGAL REPRESENTATIVE:

Wideberg, Olle Sven et al (39494), Ericsson Telecom AB, IPR Management
and Patent Department, 126 25 Stockholm, (SE)

PATENT (CC, No, Kind, Date): EP 648353 A1 950419 (Basic)
EP 648353 B1 990915
WO 9401819 940120

APPLICATION (CC, No, Date): EP 93915035 930511; WO 93SE417 930511

PRIORITY (CC, No, Date): US 907294 920701

DESIGNATED STATES: CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; NL; SE

INTERNATIONAL PATENT CLASS: G06F-009/44; G06F-009/445; G06F-009/46

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9937	3913
CLAIMS B	(German)	9937	3672

CLAIMS B	(French)	9937	4137
SPEC B	(English)	9937	10301
Total word count - document A			0
Total word count - document B			22023
Total word count - documents A + B			22023

...SPECIFICATION operations from the client-class in order to ensure that calls are transferred to the software implementation residing in the server-class. Therefore, all code **relating** to the dynamic **binding** function is found in the client- **class** .

Class declarations control the manner in which the **compiler** will store the addresses in the objects-data and in what order the addresses in the operations tables will be set forth. Some class declarations...

14/3,K/8 (Item 8 from file: 348)
 DIALOG(R)File 348:EUROPEAN PATENTS
 (c) 2004 European Patent Office. All rts. reserv.

00614517

SYSTEM AND METHOD FOR DYNAMIC RUN-TIME BINDING OF SOFTWARE MODULES IN A COMPUTER SYSTEM

SYSTEM UND VERFAHREN ZUR DYNAMISCHEN LAUFZEIT-BINDUNG VON SOFTWARE-MODULEN IN EINEM RECHNERSYSTEM.

SYSTEME ET METHODE POUR LA LIAISON DYNAMIQUE D'EXECUTION DE MODULES LOGICIELS DANS UN SYSTEME INFORMATIQUE

PATENT ASSIGNEE:

TELEFONAKTIEBOLAGET LM ERICSSON, (213761), , 126 25 Stockholm, (SE),
 (Proprietor designated states: all)

INVENTOR:

MARKSTROM, Ulf, Kurt, Haakan, Vanadisvagen 32, 3trp, S-113 46 Stockholm, (SE)

LUNDIN, Lars, Kenneth, Skimmelvagen 4, S-152 57 Sodertalje, (US)

LEGAL REPRESENTATIVE:

Wideberg, Olle Sven et al (39494), Ericsson Telecom AB, IPR Management and Patent Department, 126 25 Stockholm, (SE)

PATENT (CC, No, Kind, Date): EP 648352 A1 950419 (Basic)
 EP 648352 B1 991208
 WO 9401818 940120

APPLICATION (CC, No, Date): EP 93915034 930511; WO 93SE416 930511

PRIORITY (CC, No, Date): US 907307 920701

DESIGNATED STATES: CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; NL; SE

INTERNATIONAL PATENT CLASS: G06F-009/44

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9949	2198
CLAIMS B	(German)	9949	2090
CLAIMS B	(French)	9949	2648
SPEC B	(English)	9949	8272
Total word count - document A			0
Total word count - document B			15208
Total word count - documents A + B			15208

...SPECIFICATION from the client-class objects in order to ensure that calls are transferred to the software implementation residing in the server-class. Therefore, all code **relating** to the dynamic **binding** function is found in the client- **class** .

Class declarations control the manner in which the **compiler** will store the addresses in the objects-data and in what order the addresses in the operations tables will be set forth. Some class declarations...

14/3,K/11 (Item 3 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00887103 **Image available**

DATA SOURCE INTEGRATION SYSTEM AND METHOD
SYSTEME ET PROCEDE D'INTEGRATION DE SOURCES DE DONNEES

Patent Applicant/Assignee:

THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, 1111 Franklin Street, 5th
floor, Oakland, CA 94607-5200, US, US (Residence), US (Nationality),
(For all designated states except: US)

Patent Applicant/Inventor:

GUPTA Amarnath, 12510 Carmel Creek Road, Apt. 199, San Diego, CA 92130,
US, US (Residence), IN (Nationality), (Designated only for: US)
LUDAESCHER Bertram, 4128-148 Via Candidiz, San Diego, CA 92130-3160, US,
US (Residence), DE (Nationality), (Designated only for: US)
MARTONE Maryann E, 8368G Via Sonoma, La Jolla, CA 92037, US, US
(Residence), US (Nationality), (Designated only for: US)

Legal Representative:

FITZSIMONS Thomas R (agent), Greer, Burns & Crain, Ltd., Suite 2500, 300
South Wacker Drive, Chicago, IL 60606, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200221259 A1 20020314 (WO 0221259)
Application: WO 2001US27653 20010907 (PCT/WO US0127653)
Priority Application: US 2000231094 20000908

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ
DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ
LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG
SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 19247

Fulltext Availability:

Detailed Description

Detailed Description

... fuel lines), for
instance, specifies that for the class fuel
pump, the method "supplies" produces
result of the type "fuel-lines". The relation (R, A1/ C1 ,A,,/CJ)
specifies an n-ary **relationship** between objects of **classes** C1,- -
-Cn, with An **corresponding** to attributes or **association** roles.
In addition to these logic **declarations** that logically relate data from
different databases, GCM's 218-222 further preferably comprise additional
logic rules referred to herein as integrity constraints. The integrity...

14/3,K/12 (Item 4 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00801702 **Image available**

SYSTEM AND METHOD SUPPORTING PLURAL OPTION DATA STRUCTURES
SYSTEME ET PROCEDE POUR LA PRISE EN CHARGE DE PLUSIEURS STRUCTURES DE
DONNEES D'OPTIONS

Patent Applicant/Assignee:

CURL CORPORATION, 8th floor, 400 Technology Square, Cambridge, MA 02139,
US, US (Residence), US (Nationality)

Inventor(s):

HALSTEAD Robert H Jr, 24 Louise Road, Belmont, MA 02478, US,
KRANZ David A, 115 High Haith Road, Arlington, MA 02476, US,
TERMAN Christopher J, 60 Cedar Street, Newton Center, MA 02459, US,
WARD Stephen A, 199 Coolidge Avenue #803, Watertown, MA 02472, US,

Legal Representative:

SMITH James M (et al) (agent), Hamilton, Brook, Smith & Reynolds, P.C.,
530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200135214 A2-A3 20010517 (WO 0135214)

Application: WO 2000US30021 20001031 (PCT/WO US0030021)

Priority Application: US 99162825 19991101; US 2000672564 20000928

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE

(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 13405

Fulltext Availability:

Detailed Description

Detailed Description

... unset c.x becomes fc.remove-option <x>l

fc.set? c.xj becomes fc.option-set? <x>l

The notation <x> represents an "option **binding**" generated by the **compiler** that serves as a key for option storage and lookup. Each **class** that contains an option **declaration** for x has an **associated** option **binding** for x. When the **compiler** 1 5 **compiles** a reference such as c.x, where c is an instance of a class C, the **compiler** locates the most specific option binding for x that is in the class C or one of its superclasses. The lookup algorithm used for this process is exactly the same as is used when **compiling** references to fields. Thus, if C itself contains a declaration of the option x, the **binding** **associated** with that **declaration** is used. If not, then if C's 20 base **class** D has such a **declaration**, D's **binding** is used. If not, the algorithm proceeds in the same fashion until a declaration of x is located. The term "most ific option binding" is...that is reached by following the superclass-option-binding links repeatedly until the last OptionBindinor in the chain is reached

next: used to build a **linked** list of the OptionBindiners that **correspond** to

declarations in a particular **class**

The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindiners includinOrbindinCFs 1 5 30 and 32. Option...Ignoring other classes not relevant to this discussion, the resulting class hierarchy looks like Figure 8.

1 0 If, say, the definition of the Rectangle **class** contains a **declaration** such as

(nonlocal-option background: any="white"

an option **binding** for background will also be created in the class GraphicOptions, which means that the background option will be able to be set not just on a Rectangle but also on a VBox, HBox, or any other Graphic.

1 5 Consider the following code whose **compilation** produces the data structure of Figure 9.

ldefine-class public GraphicOptions linherits OptionListj

l

We can then define the following subclasses.

```
{define-class public Visual...
```

14/3,K/13 (Item 5 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00799786 **Image available**

SYSTEM AND METHOD SUPPORTING MAPPING OF OPTION BINDINGS

SYSTEME ET PROCEDE PERMETTANT DE SUPPORTER UNE MISE EN CORRESPONDANCE DE
LIAISONS D'OPTION

Patent Applicant/Assignee:

CURL CORPORATION, 8th floor, 400 Technology Square, Cambridge, MA 02139,
US, US (Residence), US (Nationality)

Inventor(s):

HALSTEAD Robert H Jr, 24 Louise Road, Belmont, MA 02478, US,
KRANZ David A, 115 High Haith Road, Arlington, MA 02476, US,
TERMAN Christopher J, 60 Cedar Street, Newton Center, MA 02459, US,
WARD Stephen A, 199 Coolidge Avenue, # 803, Watertown, MA 02472, US,

Legal Representative:

SMITH James M (et al) (agent), Hamilton, Brook, Smith & Reynolds, P.C.,
530 Virginia Road, P.O. Box 9133, Concord, Massachusetts, 01742-9133,
US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200133348 A2-A3 20010510 (WO 0133348)

Application: WO 2000US29899 20001030 (PCT/WO US0029899)

Priority Application: US 99162825 19991101; US 2000672579 20000928

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE

(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 13093

Fulltext Availability:

Detailed Description

Detailed Description

... unset c.x becomes fc.remove-option <x>l

fls-set? c.x) becomes fc.option-set? <x>l

The notation <x> represents an "option **binding** " generated by the **compiler** that serves as a key for option storage and lookup. Each **class** that contains an option **declaration** for x has an **associated** option **binding** for x. When the **compiler** **compiles** a reference such as c.x, where c is an instance of a class C, the **compiler** locates the most specific option binding for x that is in the class C or one of its superclasses. The lookup algorithm used for this process is exactly the same as is used when **compiling** references to fields. Thus, if C itself contains a declaration of the option x, the **binding** **associated** with that **declaration** is used. If not, then if C's 1 5 base **class** D has such a **declaration** , D's **binding** is used. If not, the algorithm proceeds in the same fashion until a declaration of x is located. The term "most specific option binding" is...that is reached by following the superclass-option-binding links repeatedly until the last OptionBinding in the chain

is reached

next: used to build a **linked** list of the OptionBindings that

correspond to

declarations in a particular **class**

The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the

OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindings including bindings 30 and 32. Option binding...time the value of an option changes. The straightforward implementation of this operation is a potentially expensive search of the data structures generated by the **compiler** when the classes in question were **compiled**, but there is a shortcut. A hash table, indexed by the option name and the class of the object on which the option is being changed, can quickly yield the right option **binding**. When (as currently the case the Curl language) option **declarations** can occur only within **class** definitions, this strategy is safe because options on a **class** can only be set after instances of the class are created, which in turn can only happen after the class definition has finished being **compiled**.

Once a class definition has been **compiled**, the mapping between option names and their most specific option bindings for that class can never change, so there is no risk of the information...is a superclass of Visual, which is in turn a superclass of every graphical class. Ignoring other classes not relevant to this discussion, the resulting **class** hierarchy looks like Figure 5.

If, say, the definition of the Rectangle **class** contains a **declaration** such as
fnonlocal-option background:any="white")
an option **binding** for background will also be created in the class GraphicOptions, which means that the background option will be able to be set not just on a Rectangle but also on a VBox, HBox, or any other Graphic.

Consider the following code whose **compilation** produces the data structure of Figure 9.

```
{define-class public GraphicOptions f inherits OptionList}
I
1 5 We can then define the following subclasses.

define...
```

14/3,K/14 (Item 6 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00799785 **Image available**

SYSTEM AND METHOD SUPPORTING NONLOCAL VALUES

SYSTEME ET PROCEDE DE SUPPORT DE VALEURS NON LOCALES

Patent Applicant/Assignee:

CURL CORPORATION, 8th floor, 400 Technology Square, Cambridge, MA 02139,
US, US (Residence), US (Nationality)

Inventor(s):

HALSTEAD Robert H Jr, 24 Louise Road, Belmont, MA 02478, US,
ERANZ David A, 115 High Haith Road, Arlington, MA 02476, US,
TERMAN Christopher J, 60 Cedar Street, Newton Center, MA 02459, US,
WARD Stephen A, 199 Coolidge Avenue #803, Watertown, MA 02472, US,

Legal Representative:

SMITH James M (et al) (agent), Hamilton, Brook, Smith & Reynolds, P.C.,
530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200133347 A2-A3 20010510 (WO 0133347)

Application: WO 2000US29861 20001031 (PCT/WO US0029861)

Priority Application: US 99162825 19991101; US 2000672565 20000928

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English
Filing Language: English
Fulltext Word Count: 14080

Fulltext Availability:
Detailed Description

Detailed Description

... unset c.x becomes {c.remove-option <x>1
(is-set? c.x) becomes fc.option-set? <x>1
The notation <x> represents an "option **binding** " generated by the **compiler** that serves as a key for option storage and lookup. Each **class** that contains an option **declaration** for x has an **associated** option **binding** for x. When the **compiler** **compiles** a reference such as c.x, where c is an instance of a class C, the **compiler** locates the most specific option binding for x that is in the class C or one of its superclasses. The lookup algorithm used for this process is exactly the same as is used when **compiling** references to fields. Thus, if C itself contains a declaration of the option x, the **binding associated** with that **declaration** is used. If not, then if C's base **class** D has such a **declaration**, D's **binding** is used. If not, the algorithm proceeds in the same fashion until a declaration of x is located. The term "most specific option binding" is...that is reached by following the superclass-option-binding links repeatedly until the last OptionBinding in the chain is reached
next: used to build a **linked** list of the OptionBindings that **correspond** to **declarations** in a particular **class**
The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindings including bindings 30 and 32. Option binding...is a superclass of Visual, which is in turn a superclass of every graphical class. Ignoring other classes not relevant to this discussion, the resulting **class** hierarchy looks like Figure 8.

If, say, the definition of the Rectangle **class** contains a **declaration** such as
(nonlocal-option background:any--"white")
an option **binding** for background will also be created in the class GraphicOptions, which means that the background option will be able to be set not just on a Rectangle but also on a VBox, HBox, or any other Graphic.

Consider the following code whose **compilation** produces the data structure of Figure 9.

```
{define-class public GraphicOptions finherits OptionList)  
We can then define the following subclasses.  
  
{define-class public Visual finherits...
```

14/3,K/15 (Item 7 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00799784 **Image available**
SYSTEM AND METHOD SUPPORTING TYPE CHECKING OF OPTIONS

SYSTEME ET PROCEDE DE CONTROLE DU TYPE D'OPTIONS

Patent Applicant/Assignee:

CURL CORPORATION, 8th floor, 400 Technology Square, Cambridge, MA 02139,
US, US (Residence), US (Nationality)

Inventor(s):

HALSTEAD Robert H Jr, 24 Louise Road, Belmont, MA 02478, US,
KRANZ David A, 115 High Haith Road, Arlington, MA 02476, US,
TERMAN Christopher J, 60 Cedar Street, Newton Center, MA 02459, US,
WARD Stephen A, 199 Coolidge Avenue #803, Watertown, MA 02472, US,

Legal Representative:

SMITH James M (et al) (agent), Hamilton, Brook, Smith & Reynolds, P.C.,
530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200133346 A2-A3 20010510 (WO 0133346)
Application: WO 2000US29853 20001031 (PCT/WO US0029853)
Priority Application: US 99162825 19991101; US 2000672848 20000928

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE

(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 12836

Fulltext Availability:

Detailed Description

Detailed Description

... 171

unset c.x becomes fc.remove-option <x>)

(is-set? c.x) becomes {c.option-set? <x>}

The notation <x> represents an "option **binding** " generated by the **compiler** that serves as a key for option storage and lookup. Each **class** that contains an option **declaration** for x has an **associated** option **binding** for x. When the **compiler** compiles a reference such as c.x, where c is an instance of a class C, the **compiler** locates the most specific option binding for x that is in the class C or one of its superclasses. The lookup algorithm used for this process is exactly the same as is used when **compiling** references to fields. Thus, if C itself contains a declaration of the option x, the **binding** **associated** with that **declaration** is used. If not, then if C's base **class** D has such a **declaration**, D's **binding** is used. If not, the algorithm proceeds in the same fashion until a declaration of x is located. The term "most specific option binding" is...that is reached by following the superclass-option-binding links repeatedly until the last OptionBinding in the chain is reached

next: used to build a **linked** list of the OptionBindings that **correspond** to

declarations in a particular **class**

The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindings including bindings 30 and 32. Option binding...is a superclass of Visual, which is in turn a superclass of every graphical class. Ignoring other classes not relevant to this discussion, the resulting **class** hierarchy looks like Figure 8.

If, say, the definition of the Rectangle **class** contains a **declaration** such as

Inonlocal-option background:any--"white")
an option **binding** for background will also be created in the class
GraphicOptions, which means that the background option will be able to be
set not just on a Rectangle but also on a VJBox, HBox, or any other
Graphic.

Consider the following code whose **compilation** produces the data
structure of Figure 9.

```
Idefine-class public GraphicOptions f inherits OptionList)
```

I

We can then define the following subclasses.

```
define-class public...
```

14/3,K/16 (Item 8 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00799776 **Image available**

SYSTEM AND METHOD SUPPORTING PROPERTY VALUES AS OPTIONS

SYSTEME ET PROCEDE ACCEPTANT DES VALEURS DE PROPRIETE SOUS FORME D'OPTIONS

Patent Applicant/Assignee:

CURL CORPORATION, 8th Floor, 400 Technology Square, Cambridge, MA 02139,
US, US (Residence), US (Nationality)

Inventor(s):

HALSTEAD Robert H Jr, 24 Louise Road, Belmont, MA 02478, US,
KRANZ David A, 115 High Haith Road, Arlington, MA 02476, US,
TERMAN Christopher J, 60 Cedar Street, Newton Center, MA 02459, US,
WARD Stephen A, 199 Coolidge Avenue, #803, Watertown, MA 02472, US,

Legal Representative:

SMITH James M (et al) (agent), Hamilton, Brook, Smith & Reynolds, P.C.,
530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200133337 A2-A3 20010510 (WO 0133337)

Application: WO 2000US29907 20001030 (PCT/WO US0029907)

Priority Application: US 99162825 19991101; US 2000672562 20000928

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE

(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 12709

Fulltext Availability:

Detailed Description

Detailed Description

... unset c.x becomes fc.remove-option <x>l

fis-set? c.xj becomes {c.option-set? <x>l

The notation <x> represents an "option **binding** " generated by the
compiler that serves as a key for option storage and lookup. Each **class**
that contains an option **declaration** for x has an **associated** option
binding for x. When the **compiler** compiles a reference such as c.x,
where c is an instance of a class C, the **compiler** locates the most
specific option binding for x that is in the class C or one of its
superclasses. The lookup algorithm used for this process is exactly the
same as is used when **compiling** references to fields. Thus, if C itself
contains a declaration of the option x, the **binding** **associated** with
that **declaration** is used. If not, then if C's base **class** D has such a
declaration, D's **binding** is used. If not, the algorithm proceeds in
the same fashion until a declaration of x is located. The term "most

specific option binding" is...that is reached by following the superclass-option-binding links repeatedly until the last OptionBinding in the chain is reached
 next: used to build a **linked** list of the OptionBindings that **correspond** to **declarations** in a particular **class**
 The **compiler** also generates an object 36, 38, 40 to describe each class (e.g., A, B, and C); each of these objects points to a list of all the OptionBinding objects **corresponding** to option **declarations** appearing in the definition of that **class**.

More specifically, **class** type object 36 for **class** A points to its base class OptionList. It also points to a linked list of its option bindings including bindings 30 and 32. Option binding...is a superclass of Visual, which is in turn a superclass of every graphical class. Ignoring other classes not relevant to this discussion, the resulting **class** hierarchy looks like Figure 8.

If, say, the definition of the Rectangle **class** contains a **declaration** such as
 Inonlocal-option background: any="white"}
 an option **binding** for background will also be created in the class GraphicOptions, which means that the background option will be able to be set not just on a Rectangle but also on a VBox, HBox, or any other Graphic.

Consider the following code whose **compilation** produces the data structure of Figure 9.

```

Idefine-class public GraphicOptions {inherits OptionList}
We can then define the following subclasses.

```

```

{define-class public Visual {inherits...

```

14/3,K/17 (Item 9 from file: 349)
 DIALOG(R)File 349:PCT FULLTEXT
 (c) 2004 WIPO/Univentio. All rts. reserv.

00560516 **Image available**
METHOD AND SYSTEM FOR COMPILING SOURCE CODE CONTAINING NATURAL LANGUAGE INSTRUCTIONS
PROCEDE ET SYSTEME DE COMPILATION DE CODE DE SOURCE CONTENANT DES RESSOURCES EN LANGAGE NATUREL
 Patent Applicant/Assignee:
 KLEIN Peter A,
 Inventor(s):
 KLEIN Peter A,
 Patent Applicant/Inventor:
 KLEIN Peter A, 29 Forty-Eighth Street, Weehawken, NJ 07087, US, US
 (Residence), US (Nationality)
 Legal Representative:
 OSTROW Seth H (agent), Brown Raysman Millstein Felder & Steiner, LLP, 120 West Forty-Fifth Street, New York, NY 10036, US,
 Patent and Priority Information (Country, Number, Date):
 Patent: WO 200023889 A1 20000427 (WO 0023889)
 Application: WO 99US24090 19991015 (PCT/WO US9924090)
 Priority Application: US 98174527 19981016
 Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES
 FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU
 LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA
 UG UZ VN YU ZW
 (EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
 (OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
 (AP) GH GM KE LS MW SD SL SZ TZ UG ZW
 (EA) AM AZ BY KG KZ MD RU TJ TM
 Publication Language: English

Filing Language: English
Fulltext Word Count: 4737

Fulltext Availability:
Claims

Claim

... cannot be associated with at least one component.

22 A memory for storing data for access by a program being executed on a computer for **compiling** a source code file, the source code file comprising natural language instructions, the memory comprising:
a data structure stored in the memory, the data structure including data used by
the program, including:
a plurality of components, the components comprising objects, modules, and **classes** ;
a plurality of natural language instruction **declarations** , and
a plurality of pointers **linking** the components and the natural language instruction declarations such that each component is linked to one or more natural language instruction declarations;
the data structure...

14/3,K/18 (Item 10 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00532081 **Image available**

REFERENCING A METHOD IN OBJECT-BASED PROGRAMMING
REFERENCE D'UN PROCEDE EN PROGRAMMATION BASEE OBJET

Patent Applicant/Assignee:

MICROSOFT CORPORATION,

Inventor(s):

GOLDE Peter H,
HEJLSBERG Anders,
ROYAL Chad W,
SHARPE Tracy C,
TOUTONGHI Michael J,
WAYT Edward H,
WILTAMUTH Scott M,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9963433 A1 19991209

Application: WO 99US12299 19990603 (PCT/WO US9912299)

Priority Application: US 9889619 19980603

Designated States: CN JP AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT
SE

Publication Language: English

Fulltext Word Count: 11724

Fulltext Availability:

Detailed Description

Detailed Description

... this search. Whether a method declaration is accessible to a method invocation depends on the access modifier (public, none, protected, or private) in the method **declaration** and on where the method designator appears. If the **class** or interface has no method **declaration** that is both **matching** and accessible, then a **compile** -time error is generated.

Compile -Time Step 3: Is the Chosen Method Appropriate?.

If there is a matching and accessible method declaration for a delegate instantiation expression, it is called the **compile** -time declaration for the delegate instantiation expression. Several further checks must be made on the compiletime declaration. A compile time error is generated if.

) If...the class to search is the one named by the TypeName. If TypeName is the name of an interface rather than a class, then a **compile** -time error occurs, because this form can designate only static methods and interfaces have no static methods. It is also necessary to determine, as in the case of wrapping instance members of a **class** , to determine if the method is appropriate. If there is a **matching** and accessible method **declaration** for a delegate instantiation expression, it is called the **compile** -time declaration for the delegate instantiation expression.

Other checks must also be made on the **compile** -time declaration. If the method designator has a MethodName of the forin Identifier, and the method designator appears within a static method, a static initializer...

14/3,K/19 (Item 11 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00454231 **Image available**
METHOD AND APPARATUS FOR UPDATING AND SYNCHRONIZING INFORMATION BETWEEN A
CLIENT AND A SERVER
PROCEDE PERMETTANT UNE MISE A JOUR ET UN SYNCHRONISATION DE L'INFORMATION
ENTRE UN UTILISATEUR ET UN SERVEUR ET DISPOSITIF CORRESPONDANT
Patent Applicant/Assignee:
APPLE COMPUTER INC,
Inventor(s):
FERRIS Michael,
POPP Nicolas,
FORSTALL Scott,
D'HARCOURT Charles,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9844695 A1 19981008
Application: WO 98US6227 19980330 (PCT/WO US9806227)
Priority Application: US 97825538 19970331
Designated States: CA JP AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Publication Language: English
Fulltext Word Count: 10078

Fulltext Availability:
Detailed Description

Detailed Description
... the object's class, for example.

The WEBOBJECT tags in Table One (INPUTFIELD, BUTTON, and OUTPUTFIELD) refer (or provide a pointer) to entries in a **declarations** file that **map** the entries to object **classes** and provide **declarations** for the tag.

These **declarations** can be used to initialize instance variables of an object and provide **runtime** environment information such as object, method and variable bindings. Table Two provides an example of the contents of a declarations file that accompanies the HTML...

14/3,K/20 (Item 12 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00253665
SYSTEM FOR CHANGING SOFTWARE DURING COMPUTER OPERATION
SYSTEME PERMETTANT LE REMPLACEMENT D'UN LOGICIEL PENDANT LE FONCTIONNEMENT
D'UN ORDINATEUR
Patent Applicant/Assignee:

TELEFONAKTIEBOLAGET LM ERICSSON,
Inventor(s):

NILSSON Kjell Rickard,
MARKSTROM Ulf Kurt Hakan,
KLOFVER Jan Leif Ingemar,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9401819 A1 19940120

Application: WO 93SE417 19930511 (PCT/WO SE9300417)

Priority Application: US 92294 19920701

Designated States: AU BR FI KR NO AT BE CH DE DK ES FR GB GR IE IT LU MC NL
PT SE

Publication Language: English

Fulltext Word Count: 16615

Fulltext Availability:

Detailed Description

Detailed Description

... operations from the client-class in order
to ensure that calls are transferred to the software
implementation residing in the server-class.

Therefore, all code **relating** to the dynamic **binding**
function is found in the client- **class** .

Class declarations control the manner in which the
compiler will store the addresses in the objects-data
and in what order the addresses in the operations
tables will be set forth. Some class declarations...

14/3,K/21 (Item 13 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00253664

SYSTEM FOR DYNAMIC RUN-TIME BINDING OF SOFTWARE MODULES IN A COMPUTER
SYSTEM

SYSTEME POUR LIAISON DYNAMIQUE D'EXECUTION DE MODULES LOGICIELS DANS UN
SYSTEME INFORMATIQUE

Patent Applicant/Assignee:

TELEFONAKTIEBOLAGET LM ERICSSON,

Inventor(s):

MARKSTROM Ulf Kurt Hakan,
LUNDIN Lars Kenneth,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9401818 A1 19940120

Application: WO 93SE416 19930511 (PCT/WO SE9300416)

Priority Application: US 92307 19920701

Designated States: AU BR FI KR NO AT BE CH DE DK ES FR GB GR IE IT LU MC NL
PT SE

Publication Language: English

Fulltext Word Count: 10067

Fulltext Availability:

Detailed Description

Detailed Description

... from the
client-class objects in order to ensure that calls are
transferred to -the software implementation residing in
the server-class. Therefore, all code **relating** to the
dynamic **binding** function is found in the client- **class** .

Class declarations control the manner in which the
compiler will store the addresses in the objects-data
and in what order the addresses in the operations
tables will be set forth. Some class declarations...

File 347:JAPIO Nov 1976-2004/Feb(Updated 040607)

(c) 2004 JPO & JAPIO

File 350:Derwent WPIX 1963-2004/UD,UM &UP=200440

(c) 2004 Thomson Derwent

Set	Items	Description
S1	237918	INTERFACE? ? OR DECLARATION? ?
S2	57774	CLASS OR CLASSES OR IMPLEMENTATION? ?
S3	322	S1(10N)S2(10N)(ASSOCIAT? OR CORRELAT? OR MAP???? OR MATCH?- ?? OR RELATE? ? OR RELATING OR RELATIONSHIP? ? OR CORRESPOND? OR LINK??? OR CONNECT??? OR AFFILIAT? OR BIND???)
S4	40	(SIGNATURE OR NAME)()MATCHING OR CALLING()CONTEXT
S5	0	DEFAULT(3N)(ASSOCIATION()RULE? ?)
S6	0	OVERRIDE(3N)(ASSOCIATION()RULE? ?)
S7	11769	MSIL OR IL OR INTERMEDIATE()LANGUAGE OR BYTECODE
S8	149881	RUNTIME OR RUN()TIME OR CL OR CLS
S9	11218	COMPIL??????
S10	58	ASSOCIATION()RULE? ?
S11	29	S3 AND S4:S10
S12	26	S11 AND IC=G06F
S13	22	S12 AND AC=US/PR
S14	16	S13 AND AY=(1970:2001)/PR
S15	16	S12 AND PY=1970:2001
S16	21	S14:S15

16/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06601797 **Image available**
RUN TIME ADDITION OF INTERFACE

PUB. NO.: 2000-187594 [JP 2000187594 A]
PUBLISHED: July 04, 2000 (20000704)
INVENTOR(s): ROBIN BRIGGS
APPLICANT(s): ADOBE SYST INC
APPL. NO.: 11-355920 [JP 99355920]
FILED: December 15, 1999 (19991215)
PRIORITY: 216683 [US 98216683], US (United States of America), December
18, 1998 (19981218)
INTL CLASS: G06F-009/44

ABSTRACT

PROBLEM TO BE SOLVED: To provide a method and a device which execute technique that adds an interface to a boss object in **run time** .

SOLUTION: A boss object 110 includes one or more **interfaces** 120, and each **interface** 120 also has a **corresponding implementation class** . Generally, this technique reads a 1st boss **class** containing a 1st **interface** defined in a 1st software component, reads a 2nd **interface** defined in another 2nd software component and also **binds** the 2nd **interface** with the 1st boss **class** so that instancing of the 1st boss **class** can produce the 1st boss object including the 1st and 2nd **interfaces** in one side face.

COPYRIGHT: (C)2000,JPO

16/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06444020 **Image available**
DYNAMICALLY CONSTRUCTING SYSTEM FOR GRAPHIC USER INTERFACE, MOUNTING METHOD
IN COMPUTER FOR DYNAMICALLY CONSTRUCTING GRAPHIC USER INTERFACE, AND **RUN -**
TIME OPERATION OF GRAPHIC USER INTERFACE

PUB. NO.: 2000-029590 [JP 2000029590 A]
PUBLISHED: January 28, 2000 (20000128)
INVENTOR(s): CARTER KAPONO D
APPLICANT(s): SUN MICROSYST INC
APPL. NO.: 11-116985 [JP 99116985]
FILED: March 19, 1999 (19990319)
PRIORITY: 44918 [US 9844918], US (United States of America), March 20,
1998 (19980320)
INTL CLASS: G06F-003/00 ; G06F-003/14

ABSTRACT

PROBLEM TO BE SOLVED: To supply an application having only a requested function by allowing an execution unit to determine whether or not a command **class** is made into an instance successfully and then construct a graphic user **interface** including the additional function **corresponding** to the command **class** made into the instance successfully.

SOLUTION: A constructor when arranging all requested packages declares that instance processing is successful and also reports that the instance processing is successful to the execution unit (4310). In this response, a graphic user interface is reconstructed and one control or more are added for the command class made into the instance successfully (4320). Then, it is decided whether or not there is still a command **class** to be made into an instance (4330) and when not, a graphic user **interface** with an additional control **relating** to the command **class** made into the instance successfully is displayed (4340).

16/5/3 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015738267 **Image available**
WPI Acc No: 2003-800468/200375
XRPX Acc No: N03-641365

Object oriented components testing apparatus for visual programming systems, has visual builder binding proxy component property to another property, and transport mechanism invoking method using value of property

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: KOBAYASHI D

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6633888	B1	20031014	US 99243261	A	19990203	200375 B

Priority Applications (No Type Date): US 99243261 A 19990203

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6633888	B1	33	G06F-017/00		

Abstract (Basic): US 6633888 B1

NOVELTY - The apparatus has a parser responsive to a component class for extracting a method having a parameter, and a **compiler** for creating a proxy component from the extracted method. A visual builder binds the proxy component property to another property in the component class. A transport mechanism then invokes the method using the value of the proxy component property as the parameter.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- (1) a method of testing component class with a visual builder
- (2) a computer program product for testing component class with a visual builder.

USE - Used for testing object oriented components in visual programming systems.

ADVANTAGE - The apparatus efficiently creates, edits, **compiles** and tests newly created component **classes** within the visual builder **interface** and **associates** methods among component **classes** within the composite component **class**.

DESCRIPTION OF DRAWING(S) - The drawing shows a flowchart for generating a bean-based application using an object oriented components testing apparatus.

pp; 33 DwgNo 8/20

Index Terms: OBJECT; ORIENT; COMPONENT; TEST; APPARATUS; VISUAL; PROGRAM; SYSTEM; VISUAL; BUILD; BIND; COMPONENT; PROPERTIES; PROPERTIES; TRANSPORT; MECHANISM; INVOKE; METHOD; VALUE; PROPERTIES

Derwent Class: T01

International Patent Class (Main): G06F-017/00

File Segment: EPI

16/5/4 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015268283 **Image available**
WPI Acc No: 2003-329212/200331
XRPX Acc No: N03-263348

Computer implemented programming method for graphical interface, involves creating design-time environment for defining customized object classes and definitions so that objects are instantiated independent of environment

Patent Assignee: RUST W C (RUST-I); WORTHEM D W (WORT-I)

Inventor: RUST W C; WORTHEM D W

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020199031	A1	20021226	US 2001872608	A	20010601	200331 B

Priority Applications (No Type Date): US 2001872608 A 20010601

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 20020199031	A1	38	G06F-009/44	

Abstract (Basic): US 20020199031 A1

NOVELTY - An user request for customizing object classes defined in a model is received. A design-time environment (200) for defining the classes is created based on the input and class functionality information is retrieved to instantiate objects independent of design-time environment. The subsets of elective class functionality with respect to customized class are received through design-time environment to define class definitions.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) computer readable program product for generating customized object classes; and

(2) computer implemented customized object class designing system

USE - For creating customized object **classes** in designing graphical **interface** of computer system **connected** to LAN, WAN, internet.

ADVANTAGE - Enables creating and implementing customized object classes effectively in a graphical design environment without dependency of class instances with environment. Improves scalability and reusability of environment by simplifying isolation of object model and service from environment while integrating instances into objects.

DESCRIPTION OF DRAWING(S) - The figure shows a block diagram of customized object classes design system in design time and **runtime** environments.

design time environment (200)

pp; 38 DwgNo 2/10

Title Terms: COMPUTER; IMPLEMENT; PROGRAM; METHOD; GRAPHICAL; INTERFACE; DESIGN; TIME; ENVIRONMENT; DEFINE; CUSTOMISATION; OBJECT; CLASS; DEFINE; SO; OBJECT; INDEPENDENT; ENVIRONMENT

Derwent Class: T01

International Patent Class (Main): G06F-009/44

File Segment: EPI

16/5/5 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

015138537 **Image available**

WPI Acc No: 2003-199063/200319

XRFX Acc No: N03-158281

Java metadata interface implementation method for computer system, involves implementing package proxy, class proxy and class instance interfaces when received request associated with metamodel has corresponding requests

Patent Assignee: HREBEJK P (HREB-I); MATULA M (MATU-I)

Inventor: HREBEJK P; MATULA M

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020165786	A1	20021107	US 2001847781	A	20010501	200319 B

Priority Applications (No Type Date): US 2001847781 A 20010501

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 20020165786	A1	35	G06F-017/60	

Abstract (Basic): US 20020165786 A1

NOVELTY - A Java metadata **interface** (JMI) **implementation** request **associated** with a metamodel, is received. The metamodel has a package which includes a **class** comprising attribute, reference and operation. The package proxy, **class** proxy and **class** instance JMI **interfaces** are implemented when the request comprises the package proxy, **class** proxy and **class** instance request respectively.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is included for Java metadata interface implementation program storage device.

USE - For implementing Java metadata interfaces for computer system connected to network such as LAN, WAN, internet, fiber optic network, ATM network, frame relay network, etc.

ADVANTAGE - The implementation of JMI interface is done automatically at **run - time**, thus reducing the amount of manual coding with enhanced repository flexibility.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the client computer system.

pp; 35 DwgNo 4/22

Title Terms: INTERFACE; IMPLEMENT; METHOD; COMPUTER; SYSTEM; IMPLEMENT; PACKAGE; CLASS; CLASS; INSTANCE; INTERFACE; RECEIVE; REQUEST; ASSOCIATE; CORRESPOND; REQUEST

Derwent Class: T01

International Patent Class (Main): G06F-017/60

File Segment: EPI

16/5/6 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

014707872 **Image available**

WPI Acc No: 2002-528576/200256

XRPX Acc No: N02-418542

Digital computer programming method for e.g. ENIAC involves representing supertype and subtype values as states of classes of objects with interfaces

Patent Assignee: TAJEA CORP (TAJE-N); HILLS T S (HILL-I)

Inventor: HILLS T S

Number of Countries: 097 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200251055	A2	20020627	WO 2001US49044	A	20011218	200256 B
AU 200227434	A	20020701	AU 200227434	A	20011218	200264
US 20020138819	A1	20020926	US 2000741502	A	20001219	200265

Priority Applications (No Type Date): US 2000741502 A 20001219

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200251055 A2 E 156 H04L-000/00

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW

AU 200227434 A H04L-000/00 Based on patent WO 200251055

US 20020138819 A1 G06F-009/45

Abstract (Basic): WO 200251055 A2

NOVELTY - Data types described as abstract without default or implicit **implementations** are distinct from **classes** or **interfaces**. The subtype and supertype **relationships** between the data types are described. The supertype and subtype values are represented as states of **classes** of objects with **interfaces**.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

- (1) **Compilation** method;
- (2) Source program retargeting method;
- (3) Recorded medium storing D language program; and

(4) Signal propagated over a propagating medium.

USE - For programming digital computer e.g. ENIAC using assembly language (D language).

ADVANTAGE - Since the D language can be used as an **intermediate language** in **compilation** and as or source language for high level programming using an object-oriented approach, the intellectual burden of the **compilers** are reduced and hence they can concentrate on the allocation and optimization problems. The object-oriented programming language enables identifying software visible physical objects composing computers as pre-existing instances of the classes.

DESCRIPTION OF DRAWING(S) - The figure shows a unified modeling language (UML) diagram of the computer programming method.

pp; 156 DwgNo 1/6

Title Terms: DIGITAL; COMPUTER; PROGRAM; METHOD; REPRESENT; VALUE; STATE; CLASS; OBJECT; INTERFACE

Derwent Class: T01

International Patent Class (Main): G06F-009/45 ; H04L-000/00

File Segment: EPI

16/5/7 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

014373999 **Image available**

WPI Acc No: 2002-194702/200225

XRPX Acc No: N02-147844

Applet execution method for Java virtual machine, involves downloading applet using class interface identified by class identifiers of new Java version, to execute applets in original version

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: MARCHESSEAUULT B T

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6343308	B1	20020129	US 97911153	A	19970814	200225 B

Priority Applications (No Type Date): US 97911153 A 19970814

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 6343308	B1	11	G06F-017/00	

Abstract (Basic): US 6343308 B1

NOVELTY - An applet with objective **classes** referred within applet is downloaded to a client, through a **class interface**. The **interface** is noted with identifiers **corresponding** to each **class associated** to new Java version of Java virtual machine (JVM). The downloaded applet is executed with respect to original version without any error with new version classes.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Data processing system;

(b) Computer program product comprising computer readable medium to perform applet execution

USE - For executing applets within JVM in object oriented computer programming e.g. Java programming used for creation of animation, graphics.

ADVANTAGE - Simplifies applet execution by eliminating individual writing of Java versions in JVM. Eliminates the need to verify the version classes, as **compiling** is enabled using common class interface.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart representing applet execution routine.

pp; 11 DwgNo 3/3

Title Terms: EXECUTE; METHOD; VIRTUAL; MACHINE; CLASS; INTERFACE; IDENTIFY; CLASS; IDENTIFY; NEW; VERSION; EXECUTE; ORIGINAL; VERSION

Derwent Class: T01

International Patent Class (Main): G06F-017/00

International Patent Class (Additional): G06F-009/00
File Segment: EPI

16/5/8 (Item 6 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

013467973 **Image available**
WPI Acc No: 2000-639916/ 200062
XRPX Acc No: N00-474614

Run - time addition of an interface to extend the functionality to
computer program objects through an object manager

Patent Assignee: ADOBE SYSTEMS INC (ADOB-N); BRIGGS R (BRIG-I)

Inventor: BRIGGS R

Number of Countries: 027 Number of Patents: 005

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1016963	A2	20000705	EP 99308850	A	19991105	200062 B
JP 2000187594	A	20000704	JP 99355920	A	19991215	200062
US 20020046395	A1	20020418	US 98216683	A	19981218	200228
US 6397384	B1	20020528	US 98216683	A	19981218	200243
US 20040015944	A1	20040122	US 98216683	A	19981218	200407
			US 2002123778	A	20020415	

Priority Applications (No Type Date): US 98216683 A 19981218; US 2002123778
A 20020415

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

EP 1016963	A2	E	17	G06F-009/44	
------------	----	---	----	-------------	--

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT
LI LT LU LV MC MK NL PT RO SE SI

JP 2000187594	A		18		
---------------	---	--	----	--	--

US 20020046395	A1			G06F-009/44	
----------------	----	--	--	-------------	--

US 6397384	B1			G06F-009/45	
------------	----	--	--	-------------	--

US 20040015944	A1			G06F-009/44	Cont of application US 98216683 Cont of patent US 6397384
----------------	----	--	--	-------------	--

Abstract (Basic): EP 1016963 A2

NOVELTY - An object manager (105) adds the supplementary
interfaces (120) of an add-in class (115) to the corresponding
boss class of run - time , to extend or modify the functionality of
the running boss object (110). Each interface (120) inherits from a
common base interface class , first and second interfaces are read
and the second interface is bound to the first boss class , to
create a boss object.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for methods
of extending a boss object, adding an interface to a boss object and
for dynamically adding an interface, for a computer program on a
computer-readable medium and for a computer system.

USE - Extending functionality at run - time to a computer program
object.

DESCRIPTION OF DRAWING(S) - The drawing is a block diagram of an
object model showing interfaces and implemented objects

Object manager (105)

Supplementary interfaces (120)

Add-in class (115)

Boss object (110)

pp; 17 DwgNo 2/7

Title Terms: RUN; TIME; ADD; INTERFACE; EXTEND; FUNCTION; COMPUTER; PROGRAM
; OBJECT; THROUGH; OBJECT; MANAGE

Derwent Class: T01

International Patent Class (Main): G06F-009/44 ; G06F-009/45

File Segment: EPI

16/5/9 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

013203202 **Image available**

WPI Acc No: 2000-375075/ 200032

XRPX Acc No: N00-281653

Interface establishing system between object-oriented software library and application program, has interface class operable to expose external interface methods to application program, as virtual function

Patent Assignee: STERLING SOFTWARE INC (STER-N)

Inventor: BIRZE B B; TRAUGHER T J

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6058396	A	20000502	US 9737839	A	19970206	200032 B
			US 9815670	A	19980129	

Priority Applications (No Type Date): US 9737839 P 19970206; US 9815670 A 19980129

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 6058396	A	8	G06F-017/30	Provisional application US 9737839

Abstract (Basic): US 6058396 A

NOVELTY - The system has an interface class operable to expose the external interface methods to the application program, as virtual functions. An internal class is operable to inherit the interface class and provide implementation of the virtual functions defined in the interface class.

DETAILED DESCRIPTION - The internal class in the object-oriented software, comprises external interface methods and internal services. A factory class (42) is operable to create multiple instances of the internal class. A factory class constructor is provided to perform assertion and parameter type checking for the internal class. An INDEPENDENT CLAIM is also included for a method for providing an interface to object-oriented software library stored in the memory.

USE - For establishing interface between object-oriented software library and application program in computer system.

ADVANTAGE - Provides stable application **interface**, at the same time preserving the product designer's freedom to modify the internal architecture. **Binds** virtual abstract **interface classes** and hidden internal **class** hierarchies through multiple inheritance to preserve **compile** time checking and limit **run time** object block. Allows protection of object life cycles through controlled creation and destruction methods. Provides the benefits of C++'s strict type checking to developers and users of the product. Class developers also benefit from lower **run time** complexity, as class users can invoke an internal objects methods without the assistance of additional envelope or bridge object.

DESCRIPTION OF DRAWING(S) - The figure illustrates the evolution of interface.

Factory class (42)

pp; 8 DwgNo 8/8

Title Terms: INTERFACE; ESTABLISH; SYSTEM; OBJECT; ORIENT; SOFTWARE; LIBRARY; APPLY; PROGRAM; INTERFACE; CLASS; OPERATE; EXPOSE; EXTERNAL; INTERFACE; METHOD; APPLY; PROGRAM; VIRTUAL; FUNCTION

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

16/5/10 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

013176371 **Image available**

WPI Acc No: 2000-348244/ 200030

XRPX Acc No: N00-260811

Improving performance of container access in workflow management system

by shifting work from runtime environment to building environment

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
RD 431094	A	20000310	RD 2000431094	A	20000220	200030 B

Priority Applications (No Type Date): RD 2000431094 A 20000220

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
RD 431094	A	4	G06F-000/00	

Abstract (Basic): RD 431094 A

NOVELTY - Method is based on extracting all information from the API calls and generating an access module for the container fields. A representation is generated for each API call which allows direct access to the container fields. For example, all syntactical checking has been performed and all symbolic references are replaced by direct reference to the fields in the workflow management system internal buffers. During build time the activity source code is scanned to determine which fields are accessed, the access module is generated and stored, selected **interface** calls are replaced by calls supporting the selected **implementation**, **compiling** and **link** editing is completed and the repository is updated.

USE - Method is for workflow management of e.g. computer usage.

ADVANTAGE - Method reduces the amount of instructions needed to carry out a particular request and eliminates processing for fields not used by an activity implementation.

DESCRIPTION OF DRAWING(S) - The figure shows the workflow dimensions.

pp; 4 DwgNo 1/6

Title Terms: IMPROVE; PERFORMANCE; CONTAINER; ACCESS; MANAGEMENT; SYSTEM; SHIFT; WORK; ENVIRONMENT; BUILD; ENVIRONMENT

Derwent Class: T01

International Patent Class (Main): G06F-000/00

File Segment: EPI

16/5/11 (Item 9 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012687656 **Image available**

WPI Acc No: 1999-493765/ 199941

XRPX Acc No: N99-367826

Element manager infrastructure in Telecommunications Management Network

Patent Assignee: CROSSKEYS SYSTEMS CORP (CROS-N)

Inventor: DUIMOVICH F; KADAR V; TURNER R

Number of Countries: 021 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9929067	A1	19990610	WO 98CA1113	A	19981201	199941 B
CA 2223123	A1	19990601	CA 2223123	A	19971201	199947
EP 1038373	A1	20000927	EP 98958750	A	19981201	200048
			WO 98CA1113	A	19981201	

Priority Applications (No Type Date): CA 2223123 A 19971201

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
WO 9929067	A1	E 13	H04L-012/24	

Designated States (National): CA JP US

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE

CA 2223123 A1 E H04L-012/24

EP 1038373 A1 E H04L-012/24 Based on patent WO 9929067

Designated States (Regional): DE FR GB IT SE

Abstract (Basic): WO 9929067 A1

NOVELTY - The infrastructure has an element process having an interface to a browser. The interface is a HTTP proxy. The HTTP proxy exchanges messages with objects of the element management process. The browser in an Internet browser. An HTTP parser creates dynamic HTML pages by replacing keywords in static pages with data that has been created and **compiled** by the proxy.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is included for a method of interacting with a network element manager, and use of a web browser as a graphical user interface for a network element manager.

USE - For element management infrastructure in Telecommunications Management Network.

ADVANTAGE - Provides element Management Infrastructure in Telecommunications Management Network model having interface to browser.

DESCRIPTION OF DRAWING(S) - The figure shows the high level objects, with their **relationships** to each other, involved in the design and **implementation** of the HTTP **interface**.

pp; 13 DwgNo 1/2

Title Terms: ELEMENT; MANAGE; TELECOMMUNICATION; MANAGEMENT; NETWORK

Derwent Class: T01; W01

International Patent Class (Main): H04L-012/24

International Patent Class (Additional): G06F-009/44 ; H04L-012/56

File Segment: EPI

16/5/12 (Item 10 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012588029 **Image available**

WPI Acc No: 1999-394136/ 199933

XRPX Acc No: N99-294556

Objects implementing system for file system

Patent Assignee: NOVELL INC (NOVE-N)

Inventor: CHRISTIANSEN N R

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5915253	A	19990622	US 96766685	A	19961213	199933 B

Priority Applications (No Type Date): US 96766685 A 19961213

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 5915253	A	17	G06F-017/30	

Abstract (Basic): US 5915253 A

NOVELTY - A registration routine stores class information about registered class in a **class** structure. The **class** information is **associated** with registered **class**, including a reference to a **class** method having the **class** method **interface**. A creation routine receives request to create an object and accesses the **class** structure to obtain properties of **class** to be created.

DETAILED DESCRIPTION - The class method interface of storage system comprises class method having a predetermined prototype. The storage system is operated to invoke a class method with predetermined prototype. An INDEPENDENT CLAIM is also included for the method of implementing an object.

USE - For file system in computer operating system.

ADVANTAGE - Allows **run time** registration of new classes which define the attributes and methods of object. Provides predetermined object interface which can interact with new objects having same predetermined interface. Enables to implement new objects without recompilation of storage system.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram illustrating attributes of volume object.

pp; 17 DwgNo 6/7

Title Terms: OBJECT; IMPLEMENT; SYSTEM; FILE; SYSTEM

Derwent Class: T01

International Patent Class (Main): G06F-017/30
File Segment: EPI

16/5/13 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

012587867 **Image available**
WPI Acc No: 1999-393974/ 199933
XRPX Acc No: N99-294398

Object oriented emitter framework for use in data processing system
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: ACKER L E; CONNER M H; MARTIN A R
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
US 5911076 A 19990608 US 9377350 A 19930614 199933 B

Priority Applications (No Type Date): US 9377350 A 19930614

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes
US 5911076 A 27 G06F-009/44

Abstract (Basic): US 5911076 A

NOVELTY - First intermediate data structure is translated to second data structure by creating instances from set of entry **classes** (116) **corresponding** to elements in **interface** definition file. The emitter **class** in instances is sub-classed to create emitter sub- **class** . The second data structure is translated to desired output file to create new instance of emitter sub-class.

DETAILED DESCRIPTION - The emitter class is sub-classed according to user input. A user modifiable template definition (124) describes a content and new format for section to format output of new emitter object instance. INDEPENDENT CLAIMS are also included for the following:

- (a) object oriented **compiling** method;
- (b) object oriented computer program product

USE - For creating new emitter for **compiler** and for different output file production in data processing system.

ADVANTAGE - Divides control language of **compiler** into easily maintainable and reusable components.

DESCRIPTION OF DRAWING(S) - The figure illustrate structure of emitter.

Entry classes (116)

Modifiable template definition (124)

pp; 27 DwgNo 4/13

Title Terms: OBJECT; ORIENT; EMITTER; FRAMEWORK; DATA; PROCESS; SYSTEM

Derwent Class: T01

International Patent Class (Main): G06F-009/44

File Segment: EPI

16/5/14 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

012587707 **Image available**
WPI Acc No: 1999-393814/ 199933
XRPX Acc No: N99-294248

Computer program for management of interface and generated classes of application software program - has interface object model and interface object user interface , linking together generated classes , interfaces and metadata of component.

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week

RD 422091 A 19990610 RD 99422091 A 19990520 199933 B

Priority Applications (No Type Date): RD 99422091 A 19990520

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
RD 422091	A		3	G06F-000/00	

Abstract (Basic): RD 422091 A

NOVELTY - The program generates an **interface** object that comprises the **interfaces**, generated **classes** and metadata of the component. The program then forms **associations** between the generated **classes** and the **interfaces**, automatically, when an **interface** is **compiled**. A user is allowed to persist an **association** by versioning the **association**, and a user may modify the **interface** definitions and generated **classes**, and to persist those modifications.

USE - For integrated software environment.

ADVANTAGE - The program may operate on a standalone basis, or be integrated into the integrated software development environment.

Dwg.1/1

Title Terms: COMPUTER; PROGRAM; MANAGEMENT; INTERFACE; GENERATE; CLASS; APPLY; SOFTWARE; PROGRAM; INTERFACE; OBJECT; MODEL; INTERFACE; OBJECT; USER; INTERFACE; LINK; GENERATE; CLASS; INTERFACE; COMPONENT

Derwent Class: T01

International Patent Class (Main): G06F-000/00

File Segment: EPI

16/5/15 (Item 13 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012577745 **Image available**

WPI Acc No: 1999-383852/ 199932

Related WPI Acc No: 2001-210112

XRPX Acc No: N99-287351

Computer implemented interface reference representation method in object oriented language

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: RAMALINGAM G; SRINIVASAN H

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5907707	A	19990525	US 97782995	A	19970114	199932 B

Priority Applications (No Type Date): US 97782995 A 19970114

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5907707	A		19	G06F-009/45	

Abstract (Basic): US 5907707 A

NOVELTY - For each interface reference IRI, a first identifier which identifies an **interface** method table **corresponding** to the **interface** ID within the descriptor data for the **class** **corresponding** to the object OI and a second identifier identifying the object OI are stored.

DETAILED DESCRIPTION - For each class Ci of objects **corresponding** descriptor data, is stored in a memory. The descriptor data includes an **interface** method table for each **interface** IDi implemented by the **class** Ci comprising entries identifying the **implementation** of a method declared by the **interface** IDi and the **interface** extended by the **interface** IDi. Each **interface** reference IRI is **associated** with an interface ID and **corresponds** to a given object OI.

USE - For generation and execution of **compiled** program in computer architecture for multimedia apparatus.

ADVANTAGE - Provides efficient mechanism for method invocations for controlling the jump operations to identified address.

DESCRIPTION OF DRAWING(S) - The figure shows the pictorial illustration of data structure of interface reference representing

method.

pp; 19 DwgNo 3/9

Title Terms: COMPUTER; IMPLEMENT; INTERFACE; REFERENCE; REPRESENT; METHOD;
OBJECT; ORIENT; LANGUAGE

Derwent Class: T01

International Patent Class (Main): G06F-009/45

File Segment: EPI

16/5/16 (Item 14 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012125970 **Image available**

WPI Acc No: 1998-542882/ 199846

WPI Acc No: N98-422575

Method of generating and employing run - time generated stub to
reference object in object oriented system - by transforming at run -
time information associated with remote object into stub class which
represents remote object and implements only those interfaces
identified by interface descriptor

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: HAMILTON G; KESSLER P B; PELEGRI-LLOPART E; RIGGS R; WALDO J H;
WOLLRATH A M

Number of Countries: 019 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9844414	A1	19981008	WO 98US5665	A	19980324	199846 B
US 5999988	A	19991207	US 97829861	A	19970331	200004
EP 972241	A1	20000119	EP 98913052	A	19980324	200009
			WO 98US5665	A	19980324	
JP 2002516006	W	20020528	JP 98541724	A	19980324	200238
			WO 98US5665	A	19980324	
EP 972241	B1	20030514	EP 98913052	A	19980324	200333
			WO 98US5665	A	19980324	
DE 69814611	E	20030618	DE 614611	A	19980324	200348
			EP 98913052	A	19980324	
			WO 98US5665	A	19980324	

Priority Applications (No Type Date): US 97829861 A 19970331

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9844414 A1 E 65 G06F-009/46

Designated States (National): JP

Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LU MC
NL PT SE

US 5999988 A G06F-009/46

EP 972241 A1 E G06F-009/46 Based on patent WO 9844414

Designated States (Regional): DE FR GB NL SE

JP 2002516006 W 61 G06F-009/46 Based on patent WO 9844414

EP 972241 B1 E G06F-009/46 Based on patent WO 9844414

Designated States (Regional): DE FR GB NL SE

DE 69814611 E G06F-009/46 Based on patent EP 972241

Based on patent WO 9844414

Abstract (Basic): WO 9844414 A

The method involves receiving an object reference to the remote
object from the second virtual machine. The object reference has
information associated with the remote object. The information includes
an interface descriptor and an object handle of the remote object. The
object handle identifies the remote object and the interface descriptor
identifies an interface of the remote object implemented in the second
virtual machine. The information associated with the remote object is
transformed at **run - time** into a stub class. The stub class
represents the remote object and implements only those interfaces
identified by the interface descriptor and defined by the first virtual
machine. The stub class is instantiated and the first virtual machine
is provided with an instance associated with the stub class.

Dwg.1/19

Title Terms: METHOD; GENERATE; EMPLOY; RUN; TIME; GENERATE; STUB; REFERENCE
; OBJECT; OBJECT; ORIENT; SYSTEM; TRANSFORM; RUN; TIME; INFORMATION;
ASSOCIATE; REMOTE; OBJECT; STUB; CLASS; REPRESENT; REMOTE; OBJECT;
IMPLEMENT; INTERFACE; IDENTIFY; INTERFACE; DESCRIBE
Derwent Class: T01
International Patent Class (Main): G06F-009/46
File Segment: EPI

16/5/17 (Item 15 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

011638396 **Image available**
WPI Acc No: 1998-055304/ 199806
XRPX Acc No: N98-043808

Association mechanism data structure in distributed object computing
system - uses linking mechanism arranged to form run time link
between server rep class and selected one of servant objects at run
time

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: KESSLER P B; LIM S B; RADIA S R
Number of Countries: 020 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 817037	A2	19980107	EP 97304147	A	19970613	199806 B
JP 10091444	A	19980410	JP 97170768	A	19970626	199825
US 6044224	A	20000328	US 96670682	A	19960626	200023

Priority Applications (No Type Date): US 96670682 A 19960626

Cited Patents: No-SR.Pub

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

EP 817037	A2	E	26	G06F-009/46	
-----------	----	---	----	-------------	--

Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE

JP 10091444	A	23	G06F-009/44
-------------	---	----	-------------

US 6044224	A		G06F-015/00
------------	---	--	-------------

Abstract (Basic): EP 817037 A

The data structure includes at least one server rep class
representing a single one of the number of subcontracts. The server rep
class is arranged to provide functionality unique to the single one of
the number of subcontracts. At least one servant base class is arranged
to be associated with servant objects that are created at run time .
A linking mechanism is arranged to form a run time link between the
server rep class and a selected one of the servant objects at run
time such that the servant object may use the functionality provided
by the single one of the number of subcontracts.

At run time a server rep object associated with the server rep
class is created and the run time link includes a location
indicator from the selected servant object to the server rep object.

USE - For associating services provided by distributed object
system with object run time .

ADVANTAGE - Allows dynamically associating subcontract with
individual servant at run time instead of servant classes or
interface at compile time. Such mechanism may be also compatible
with various of object life cycle functions.

Dwg.3/13

Title Terms: ASSOCIATE; MECHANISM; DATA; STRUCTURE; DISTRIBUTE; OBJECT;
COMPUTATION; SYSTEM; LINK; MECHANISM; ARRANGE; FORM; RUN; TIME; LINK;
SERVE; CLASS; SELECT; ONE; OBJECT; RUN; TIME

Derwent Class: T01

International Patent Class (Main): G06F-009/44 ; G06F-009/46 ;
G06F-015/00

File Segment: EPI

16/5/18 (Item 16 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

011281278 **Image available**
WPI Acc No: 1997-259182/ 199723
XRPX Acc No: N97-214272

**First and second object instance dynamic linking system for OOL -
includes semantic links relating first and second object instances
via their class interfaces using dynamic binding and probing**

Patent Assignee: VIEWSOFT INC (VIEW-N)

Inventor: ANDERSON K L; MITCHELL D C; MITCHELL D K; OSMAN A V

Number of Countries: 020 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9715883	A1	19970501	WO 96US16927	A	19961022	199723 B
EP 857331	A1	19980812	EP 96936844	A	19961022	199836
			WO 96US16927	A	19961022	
US 5872973	A	19990216	US 95548536	A	19951026	199914
JP 11514114	W	19991130	WO 96US16927	A	19961022	200007
			JP 97516726	A	19961022	
EP 857331	B1	20000726	EP 96936844	A	19961022	200036
			WO 96US16927	A	19961022	
DE 69609516	E	20000831	DE 609516	A	19961022	200050
			EP 96936844	A	19961022	
			WO 96US16927	A	19961022	

Priority Applications (No Type Date): US 95548536 A 19951026

Cited Patents: 2.Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9715883	A1	E	85	G06F-009/46	
				Designated States (National): JP	
				Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE	
EP 857331	A1	E		G06F-009/46	Based on patent WO 9715883
				Designated States (Regional): AT BE CH DE FR GB IT LI	
US 5872973	A			G06F-009/44	
JP 11514114	W		78	G06F-009/44	Based on patent WO 9715883
EP 857331	B1	E		G06F-009/46	Based on patent WO 9715883
				Designated States (Regional): AT BE CH DE FR GB IT LI	
DE 69609516	E			G06F-009/46	Based on patent EP 857331
					Based on patent WO 9715883

Abstract (Basic): WO 9715883 A

The dynamic object linking system includes at least one semantic link relating the first and second objects which are instances of the same object, through their class interfaces using their dynamic binding and probing capabilities. Numerous semantic links and can be connected by naming a list of connected links. The connection specification is stored. A unit creates and initialises the connection during program execution in accordance with the specification.

A third object is included with a unit to maintain the connection during program execution by changing the link to related the first and third objects when a characteristic of the first or second object changes. The connection can be destroyed during program execution. The link transforms on information passed between the first and second objects.

ADVANTAGE - Provides dynamically reconfigurable links with transformations between objects at run - time without requiring them to become dependent upon or have knowledge of each other. Provides visual programming environment supporting dynamic linkages of objects in languages supporting meta-data, full dynamic binding, probing and generic factory method capabilities.

Dwg.2/4

Title Terms: FIRST; SECOND; OBJECT; INSTANCE; DYNAMIC; LINK; SYSTEM; LINK;
RELATED; FIRST; SECOND; OBJECT; INSTANCE; CLASS; INTERFACE; DYNAMIC; BIND

; PROBE
Derwent Class: T01
International Patent Class (Main): G06F-009/44 ; G06F-009/46
File Segment: EPI

16/5/19 (Item 17 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

010938917 **Image available**
WPI Acc No: 1996-435867/ 199644

Distributed object creation for distributed object system - has wrapper
classes in distributed objects inheriting object attributes via
inheritance relationship with developer written servant class of
objects, classes inherit attributes through optional relationships
with interface class

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: BALICK M; BRACHO R; HAPNER M W; MCCHESENEY R J; SNYDER A; VAN HOFF
A A; MC CHESNEY R J
Number of Countries: 008 Number of Patents: 007

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 735474	A2	19961002	EP 96301250	A	19960223	199644 B
CA 2172423	A	19961001	CA 2172423	A	19960322	199705
EP 735474	A3	19970115	EP 96301250	A	19960223	199713
JP 9054685	A	19970225	JP 9677771	A	19960329	199718
EP 735474	B1	20030507	EP 96301250	A	19960223	200333
DE 69627926	E	20030612	DE 627926	A	19960223	200346
			EP 96301250	A	19960223	
US 6640255	B1	20031028	US 95414240	A	19950331	200372

Priority Applications (No Type Date): US 95414240 A 19950331

Cited Patents: 3.Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 735474	A2	E	28	G06F-009/46	
Designated States (Regional): DE FR GB IT SE					
CA 2172423	A			G06F-009/44	
EP 735474	A3			G06F-009/46	
JP 9054685	A		21	G06F-009/06	
EP 735474	B1	E		G06F-009/46	
Designated States (Regional): DE FR GB IT SE					
DE 69627926	E			G06F-009/46	Based on patent EP 735474
US 6640255	B1			G06F-009/44	

Abstract (Basic): EP 735474 A

The method involves providing interface files describing interfaces contained in the distributed object, and implementation files describing object implementations. Servant class files describe a servant class of object implementations. The interface, under computer control, compiles the files to produce compiled header and source files corresp. to the files.

The corresp. implementation files include template source files. Under computer control the header and source files are linked with a wrapper class containing functions and services for operating the distributed object on the system, so that the wrapper class has an inheritance relationship w.r.t. the servant class. The wrapper class is derived from the servant class to create the distributed object.

ADVANTAGE - Can be installed without extensive modification to objects and without programmer having familiarity with support provision for distributed object systems in programming code of their objects.

Dwg.9/12

Title Terms: DISTRIBUTE; OBJECT; CREATION; DISTRIBUTE; OBJECT; SYSTEM; WRAP
; CLASS; DISTRIBUTE; OBJECT; OBJECT; ATTRIBUTE; RELATED; DEVELOP; WRITING
; CLASS; OBJECT; CLASS; ATTRIBUTE; THROUGH; OPTION; RELATED; INTERFACE;
CLASS

Derwent Class: T01
International Patent Class (Main): G06F-009/06 ; G06F-009/44 ;
G06F-009/46
File Segment: EPI

16/5/20 (Item 18 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

008891128 **Image available**
WPI Acc No: 1992-018397/ 199203
XRPX Acc No: N92-013954

**Object orientated program for window based computers - has event drivers
in window tools which communicate with application using unmodified
language C**

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: NAYEEM I; ISLAM N
Number of Countries: 002 Number of Patents: 002
Patent Family:

Parent No	Kind	Date	Applicat No	Kind	Date	Week
FR 2661525	A	19911031	FR 915113	A	19910425	199203 B
US 5446902	A	19950829	US 90515427	A	19900427	199540
			US 9391536	A	19930714	

Priority Applications (No Type Date): US 90515427 A 19900427; US 9391536 A
19930714

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 5446902	A	17	G06F-007/00	Cont of application US 90515427

Abstract (Basic): FR 2661525 A

A window based computer system can be provided with an object orientated program using language C. A typical window based system has window tools (30) and library (28) linked to an application (32), also interface units and window server (10) (18), the whole being connected to the general system (24).

The window library provides a protocol procedure interface and the application calls the event drivers in the window tools to send window and drawing orders to the server which notifies events to the application in response to the users actions.

ADVANTAGE - Object orientated programming is obtained using language C without modification. (35pp Dwg.No.1/7)

Title Terms: OBJECT; ORIENT; PROGRAM; WINDOW; BASED; COMPUTER; EVENT; DRIVE
; WINDOW; TOOL; COMMUNICATE; APPLY; UNMODIFIED; LANGUAGE

Derwent Class: T01

International Patent Class (Main): G06F-007/00
International Patent Class (Additional): G06F-009/44
File Segment: EPI

16/5/21 (Item 19 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

008779786 **Image available**
WPI Acc No: 1991-283803/ 199139
XRPX Acc No: N91-217054

**Dynamic linking apparatus for linking of computer software components -
has query function allowing application programs to locate library
classes and create instances of derived classes**

Patent Assignee: APPLE COMPUTER INC (APPY)
Inventor: HENINGER A G
Number of Countries: 002 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
GB 2242293	A	19910925	GB 9026698	A	19901207	199139 B
US 5659751	A	19970819	US 90461386	A	19900105	199739
			US 92985449	A	19921203	

ADVANTAGE - Does not consume large portion of memory, or require powerful processor. (33pp Dwg.No.2/7


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

[Advanced Search](#)
[Preferences](#)

Web

 Results 1 - 10 of about 27,500 for compiler signature matching. (0.63 seconds)

Curiosity is bliss: The dark side of C# Delegates

... Encountering a delegate declaration, the **compiler** actually generates a class with an Invoke method that **matches** the delegate declaration's **signature**. ...

blog.monstuff.com/archives/000037.html - 20k - [Cached](#) - [Similar pages](#)

ToolTalk User's Guide

... the ptype file is compiled with the ToolTalk type **compiler**, tt_type_comp , at ... The first part of a **signature** specifies **matching** attribute values. ...

www.cs.arizona.edu/computer.help/policy/DIGITAL_unix/ToolTalkUsersGuide/tt Ug_15.html - 27k - [Cached](#) - [Similar pages](#)

[PDF] Exclusion-based Signature Matching for Intrusion Detection

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... rarely expected to **match** any **signature**, ExB performs ... an algorithm for concurrently **matching** multiple strings ... widely used in current **compiler** technology, and ...

www.ist-scampi.org/publications/papers/markatos-ccn2002.pdf - [Similar pages](#)

Static Interpretation

... Thus one can make use of the information the **compiler** has about the ... **Signature matching** generates no code either, not even in the case where **signature matching** ...

www.itu.dk/research/mikit/modules.html - 9k - [Cached](#) - [Similar pages](#)

TestEvents2

... In this code example we explicitly tell the compiler to "override" OnMouseUp, enabling the **compiler** to warn us if the **signature** does not **match** a **signature** in ...

www.geocities.com/Jeff_Louie/testevents2.htm - 15k - [Cached](#) - [Similar pages](#)

New Features In Jacl and Tcl Blend

... than once, you should think about installing and using the Jikes **compiler**. ... **Signature Matching**: The **signature matching** system introduced in Tcl/Java 1.1 did not ...

www.tcl.tk/software/java/new_features.html - 15k - [Cached](#) - [Similar pages](#)

net::nosica::compiler::typechecker::MethodResolver interface ...

... return the best **matching** method if it exists on a call context The call context is determined by : the instance; the user supplied **signature**; the accessor. In ...

www.nosica.net/old/dox/html/interfacenet_1_1nosica_1_1compiler_1_1typechecker_1_1MethodResolver.html - 49k - [Cached](#) - [Similar pages](#)

Java Technology Forums

... It's all about the way that the **compiler** decides which method to use when it ... arguments have, and the name of the function so it can find a **matching signature**. ...

forum.java.sun.com/thread.jsp?thread=526767&forum=31&message=2528137 - 24k - [Cached](#) - [Similar pages](#)

SML/NJ Module Extensions

... As usual, any attempt to **match** these inconsistent **signatures** will fail. The **compiler** flag **Compiler.Control.multDefWarn** : bool ref controls whether a warning ...

www.smlnj.org/doc/Conversion/module-extensions.html - 4k - [Cached](#) - [Similar pages](#)

415. [Andrew] late error detection in parsing 416. [dbm] equality ...
... [Zhong, Stefan, Lal, Allen] **compiler** blowup with large andalso expression 1538. ...
[dbm] open and **signature matching** with polymorphic type constructors 1551. ...
www.smlnj.org/bugs/openbugs - 15k - [Cached](#) - [Similar pages](#)
[[More results from www.smlnj.org](#)]

Google ►

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)

compiler signature matching

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

[Advanced Search](#)
[Preferences](#)

Web

 Results 1 - 10 of about **57,000** for **compiler signature matching interface class**. (0.64 seconds)

Event Handling in COM (Visual C++ Concepts)

... event handlers, which are methods with **signatures** (return types ... calling conventions do not have to **match**; see Layout ... To **compile** and run the example, refer to ...

msdn.microsoft.com/library/en-us/vccore/html/vcconEventHandlingInCOM.asp - 19k - [Cached](#) - [Similar pages](#)

Curiosity is bliss: The dark side of C# Delegates

... **Compiler**-supplied delegate methods at MSDN: Invoke, BeginInvoke and EndInvoke. Method **signature matching** isn't complete type-safety. **Interface**-style vs. ...

blog.monstuff.com/archives/000037.html - 20k - [Cached](#) - [Similar pages](#)

3.6 Signatures and overloading

... are unique within that **class**, struct, or **interface**. ... A **compile**-time error occurs if two members are ... For other purposes of **signature matching** (eg, hiding or ...

www.asprelated.com/csharp/sharp-3_6.aspx - 16k - [Cached](#) - [Similar pages](#)

Interface layout

... the effect on the various languages targeting the **compiler**, and some ... MethodImpls which **match** methods of any name that have the correct **signature**). ...

blogs.msdn.com/cbrumme/archive/2003/05/03/51381.aspx - 33k - [Cached](#) - [Similar pages](#)

Java Programmer's SourceBook : Thinking in Java

... determining if a particular **signature matches** with your ... therefore all the method **signature** information is ... that's automatically synthesized by the **compiler**. ...

www.codeguru.com/java/tij/tij0122.shtml - 43k - [Cached](#) - [Similar pages](#)

Java Programmer's SourceBook : Thinking in Java

... All derived-**class** methods that **match** the **signature** of the base-**class** declaration will be called using ... (Otherwise, the **compiler** gives you an error message.). ...

www.codeguru.com/java/tij/tij0079.shtml - 31k - [Cached](#) - [Similar pages](#)

[[More results from www.codeguru.com](#)]

JavaClass

... Returns: a java method with **matching signature**, if found. ... be passed to the **compiler**

Returns: boolean ... Specified by: getProperty in **interface** IElement Overrides ...

workshop.bea.com/extensibility/docs/en/edk/java-class/com/bea/ide/lang/java/element/JavaClass.html - 58k - [Cached](#) - [Similar pages](#)

The Code Project - Function overriding in C++/CLI - Managed C++

... method, we'd have got a **compiler** error, as ... specify which method overrides which base/**class/interface** method, provided the function **signatures match**. ...

codeproject.com/managedcpp/cpp-clioverriding.asp - 52k - [Cached](#) - [Similar pages](#)

[29] Learning C++ if you already know Smalltalk (Czê¶æ C++ FAQ ...

... For example, with static typing, the **signature matching** for function ... An improper **match** is flagged as an error by ... The C++ **compiler** exploits the added semantic ...

klub.chip.pl/b.krzemien/c++-faq-pl/smalltalk.html - 13k - [Cached](#) - [Similar pages](#)

Re: Tested compiler PR patches

Digitized by Google

compiler signature matching inter Search

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

<http://www.google.com/search?hl=en&lr=&ie=UTF-8&q=compiler+signature+matching+int...> 6/28/04



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

(compiler or parser) and (map* or match* or associat* or bind*)

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction](#)

Terms used

compiler or **parser** and **map** or **match** or **associat** or **bind** **near** **declaration** or **interface** and **implemenation** or

Sort results by

Display results

[Save results to a Binder](#)

[Search Tips](#)

☐ [Open results in a new window](#)

[Try an Advanced Search](#)

[Try this search in The ACM G](#)

Results 61 - 80 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) **[4](#)** [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance sort

61 [Ada technology development at Irvine Computer Sciences Corporation](#)

Eric W. Olsen, Stephen B. Whitehill

March 1982 **ACM SIGAda Ada Letters**, Volume 1 Issue 3

Full text available: [pdf\(372.98 KB\)](#)

Additional Information: [full citation](#)

62 [The state of the art in distributed query processing](#)

Donald Kossmann

December 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 4

Full text available: [pdf\(455.39 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Distributed data processing is becoming a reality. Businesses want to do it for many reasons, and often must do it in order to stay competitive. While much of the infrastructure for distributed data processing is already there (e.g., modern network technology), a number of issues make distributed processing still a complex undertaking: (1) distributed systems can become very large, involving thousands of heterogeneous sites including PCs and mainframe server machines; (2) the state of the art in distributed query processing is still immature.

Keywords: caching, client-server databases, database application systems, dissemination-based information systems, economic models for query processing, middleware, multitier architectures, query execution, query optimization, replication, wrappers

63 [Query-based debugging of object-oriented programs](#)

Raimondas Lencevicius, Urs Hölzle, Ambuj K. Singh

October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 32 Issue 10

Full text available: [pdf\(1.84 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Object relationships in modern software systems are becoming increasingly numerous and complex. Programmers who try to find violations of such relationships need new tools that allow them to examine objects in a large system more efficiently. Many existing debuggers present only a low-level, one-at-a-time view of objects and their relationships. We propose a new solution to overcome these problems with query-based debugging. The implementation of the query-based debugger described here offers a more efficient and comprehensive view of the system state.

64 [Timekeeping in the memory system: predicting and optimizing memory behavior](#)

Zhigang Hu, Stefanos Kaxiras, Margaret Martonosi